AREDN® Documentation

Release latest

AREDN

Jun 14, 2026

WHAT IS AREDN®

1 AREDN® Overview **3**

2 Why AREDN® **5**

3 Selecting Radio Hardware **17**

4 Downloading AREDN® Firmware **19**
4.1 Types of Firmware . . . 19
4.2 Choosing Firmware to Download . . . 19

5 Installing AREDN® Firmware **23**
5.1 Preparing your computer . . . 24
5.2 Ubiquiti 802.11n first install process . . . 25
5.3 Ubiquiti 802.11ac first install process . . . 26
5.4 Mikrotik first install process . . . 29
5.5 TP-LINK first install process . . . 35
5.6 GL-iNet first install process . . . 37
5.7 Cudy first install process . . . 37
5.8 OpenWRT One first install process . . . 39
5.9 MorseMicro device install process . . . 40
5.10 Zyxel device install process . . . 40
5.11 After the AREDN® firmware install . . . 41
5.12 Node Reset button actions . . . 41

6 Firstboot Node Setup **43**
6.1 Advanced Options . . . 44
6.2 Resetting a node to *firstboot* state . . . 45

7 Node Status Display **47**
7.1 Top Nav Bar . . . 48
7.2 Left Nav Bar . . . 49
7.3 Left Section . . . 49
7.4 Center Section . . . 50

7.5 Right Section . . . 55

8 Mesh Status Display 57

9 Node Admin Guide 59
9.1 Admin navigation & actions . . . 59
9.2 Basics . . . 60
9.3 Time settings . . . 64
9.4 Firmware settings . . . 66
9.5 Package settings . . . 71
9.6 Network settings . . . 73
9.7 Location settings . . . 81
9.8 Internal Services . . . 82
9.9 Local Services . . . 89
9.10 Local Devices . . . 93
9.11 Local Nodes . . . 94
9.12 Neighborhood Nodes . . . 95
9.13 Radios & Antennas . . . 97
9.14 Mesh section . . . 102
9.15 LAN DHCP settings . . . 103
9.16 Ethernet Ports & Xlinks . . . 106
9.17 Tunnels . . . 108
9.18 Tools . . . 113

10 Reporting Problems or Issues 123

11 Networking Overview 125

12 Network Topologies 127
12.1 Types of Topologies . . . 128
12.2 Types of Links . . . 129
12.3 Supernode Architecture . . . 131

13 Radio Spectrum Characteristics 133
13.1 5.8 GHz Characteristics . . . 135
13.2 3.4 GHz Characteristics . . . 136
13.3 2.4 GHz Characteristics . . . 136
13.4 900 MHz Characteristics . . . 137

14 Channel Planning 139
14.1 Wireless Network Operation . . . 139
14.2 Channel Plans and Frequency Coordination . . . 142
14.3 Collocated Nodes . . . 144
14.4 Channel Planning Tips . . . 148

15 Network Modeling 149

15.1 Creating a Path Profile . . . 149
15.2 Determining Node or Network Coverage . . . 153

16 AREDN® Services Overview **155**

17 Chat Programs **157**
17.1 Raven . . . 157
17.2 MeshChat . . . 158
17.3 Internet Relay Chat . . . 159
17.4 Jabber/XMPP . . . 160
17.5 Let's Chat . . . 161
17.6 Mattermost . . . 162
17.7 Matrix - Synapse . . . 163
17.8 Example Chat Service Comparison . . . 164

18 Email Programs **165**
18.1 Citadel/UX . . . 165
18.2 Open Source Email Server . . . 166
18.3 Using WinLink to Send Email . . . 167
18.4 Example Email Service Comparison . . . 168

19 File Sharing Programs **169**
19.1 FTP Services . . . 169
19.2 Web Services . . . 170
19.3 Collaborative Computing . . . 171

20 VoIP Audio/Video Conferencing **173**
20.1 VoIP Server . . . 173
20.2 VoIP Endpoints . . . 175
20.3 Video Conferencing Software . . . 177
20.4 Example VoIP Service Comparison . . . 179

21 Video Streaming and Surveillance **181**
21.1 IP Video Cameras . . . 182
21.2 Video Display Software . . . 183
21.3 Example Video Service Comparison . . . 188

22 Networking Tools **189**
22.1 Manage Extra Static Routes . . . 189
22.2 AREDN® Prometheus Exporter . . . 189
22.3 KN6PLV Mesh Map . . . 191

23 Computer Aided Dispatch **193**
23.1 EmComMap . . . 193
23.2 Open ISES Tickets . . . 194
23.3 Example Computer Aided Dispatch Comparison . . . 195

24 Other Services **197**
24.1 Network Time Services . . . 197
24.2 weeWx Weather Service . . . 198
24.3 GPS Tracking Services . . . 199

25 Beginner's Guide **203**
25.1 What it's all about . . . 203
25.2 Getting Started . . . 203
25.3 RF access to the network . . . 203
25.4 Recommended equipment . . . 204
25.5 Configuring your node . . . 206
25.6 Aiming high gain antennas . . . 207
25.7 Tools for planning your network . . . 207
25.8 Example node deployments . . . 207

26 Tips for Handling Firmware **209**
26.1 Tips for Upgrading Firmware . . . 210
26.2 Tips for Downgrading Firmware . . . 210

27 Connecting Nodes to Home Routers **213**

28 Power over Ethernet (PoE) **215**
28.1 Passive PoE . . . 215

29 Command Line Access to Your Node **217**

30 Tips for Aiming Directional Antennas **221**
30.1 Practice with Nearby Nodes . . . 221
30.2 Aligning Distant Nodes . . . 222

31 The Babel Routing Protocol **225**

32 Settings for Radio Mobile **227**

33 Comparing SISO and MIMO Hardware **229**
33.1 SISO Device Hardware . . . 230
33.2 MIMO Device Hardware . . . 230
33.3 Troubleshooting Tips . . . 231

34 Configuring a Supernode **233**
34.1 Criteria for Deploying a Supernode . . . 233
34.2 Coordinating Supernode Deployments . . . 234
34.3 Setting up a Supernode . . . 235
34.4 Configuring a Supernode Peer Tunnel . . . 237
34.5 44Net and Supernodes . . . 237

35 Creating a Local AREDN® Software Server **239**
35.1 Configure your software server . . . 239
35.2 Point nodes to the local server . . . 241

36 Using Xlinks **243**
36.1 Configure the AREDN® nodes at both ends . . . 243
36.2 Configure the intermediate transport link . . . 245

37 Custom App Launcher **247**
37.1 Adding an app badge counter and badge-color . . . 248

38 Virtual Machine Installs **249**
38.1 Prerequisites / Image information . . . 249
38.2 Proxmox Installs . . . 250
38.3 QEMU Installs . . . 250
38.4 VMware Installs . . . 250

39 Tools for Integrators **253**
39.1 SYSINFO . . . 253

40 Contributing to Documentation **259**

41 Responsible Disclosure Policy **261**

42 Frequencies and Channels **263**

43 Acronyms List **265**

44 License **279**
44.1 Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International . 279

Release
latest

This documentation set consists of several sections which are shown in the navigation list.

- The **Getting Started Guide** walks through the process of configuring an AREDN® radio node to be part of a mesh network.
- The **Network Design Guide** provides background information and tips for planning and deploying a robust mesh network.
- The **Applications and Services Guide** discusses the types of programs or services that can be used across a mesh network.
- The **How-to Guides** provide tips and techniques for various tasks.
- Finally, the **Appendix** contains supplementary information.

If you wish to locate specific topics within the documentation, you can type keywords into the *Search docs* field to display a list of items which match your search.

If you would like to see the documentation for a specific AREDN® release, click on the **Read the Docs** label at the bottom of the navigation bar. This label shows the version you are currently viewing, but clicking the label bar opens a panel with several other options. Here you may choose to view another version of the documentation, and you can also download the entire documentation set in any of several formats *(PDF, ePub, HTML)* for offline use.

Additional information about the AREDN® project can be found at the links below.

- AREDN® homepage
- AREDN® forums

Note: AREDN® is a registered trademark of *Amateur Radio Emergency Data Network, Inc.* and may not be used without permission.

CHAPTER

ONE

AREDN® OVERVIEW

The AREDN® acronym stands for "Amateur Radio Emergency Data Network" and it provides a way for *Amateur Radio* operators to create high-speed ad hoc *Data Networks* for use in *Emergency* and service-oriented communications.

For many years amateur radio operators and their served agencies have relied on voice transmissions for emergency or event communications. A typical message-passing scenario involved conveying the message to a radio operator who would write or type it onto a standard ICS-213 form. The message would then be relayed by radio to another operator who would write or type it on another ICS-213 form at the receiving end. The form would typically be hand-delivered to the recipient who would read and sign the form. Any acknowledgement or reply would then be handled through the same process from the receiving end back to the originator.

This tried-and-true scenario has worked well, and it continues to work for handling much emergency and event traffic. Today, however, digital transmission is more commonly used instead of traditional methods and procedures. The hardcopy ICS-213 form is giving way to the Winlink electronic form, with messages being passed using digital technologies such as AX.25 packet, HF Pactor, Fldigi, and others.

Our Mission

The primary goal of the AREDN® project is to empower licensed amateur radio operators to quickly and easily deploy high-speed data networks when and where they are needed.

In today's high-tech society people have become accustomed to different ways of handling their communication needs. The preferred methods involve short messaging and keyboard-to-keyboard communication, along with audio-video communication using Voice over IP (VoIP) and streaming technologies.

The amateur radio community is able to meet these high-bandwidth digital communication requirements by using FCC Part 97 amateur radio frequency bands to send digital data between devices which are linked with each other to form a self-healing, fault-tolerant data network. Some have described this as an amateur radio version of the Internet. Although it is not intended for connecting people to **the Internet**, an AREDN® mesh network will provide typical Internet or intranet-type

applications to people who need to communicate across a wide area during an emergency or community event.

An AREDN® network is able to serve as the transport mechanism for the preferred applications people rely upon to communicate with each other in the normal course of their business and social interactions, including email, chat, phone service, document sharing, video conferencing, and many other useful programs. Depending on the characteristics of the AREDN® implementation, this digital data network can operate at near-Internet speeds with many miles or kilometers between network nodes.

A foundational design goal of the AREDN® project is to minimize the technical expertise that is normally required to configure a robust radio network. Devices running AREDN® firmware are in many ways self-configuring so that users without a background in IP networking can easily build or connect to a local RF network. As mentioned in a recent Amateur Radio Digital Communications (ARDC) annual report, "AREDN® software allows volunteers to set up a node with minimal expertise and effort, and because the software configures the network automatically, advanced network technology is not needed."

This facilitates the primary goal of the AREDN® project, which is to empower licensed amateur radio operators to quickly and easily deploy high-speed data networks when and where they are needed, as a service both to the hobby and the community. This is especially important in cases when traditional "utility" services (electricity, phone lines, or Internet services) become unavailable. In those cases an off-grid amateur radio emergency data network may be a lifeline for communities impacted by a local disaster.

CHAPTER

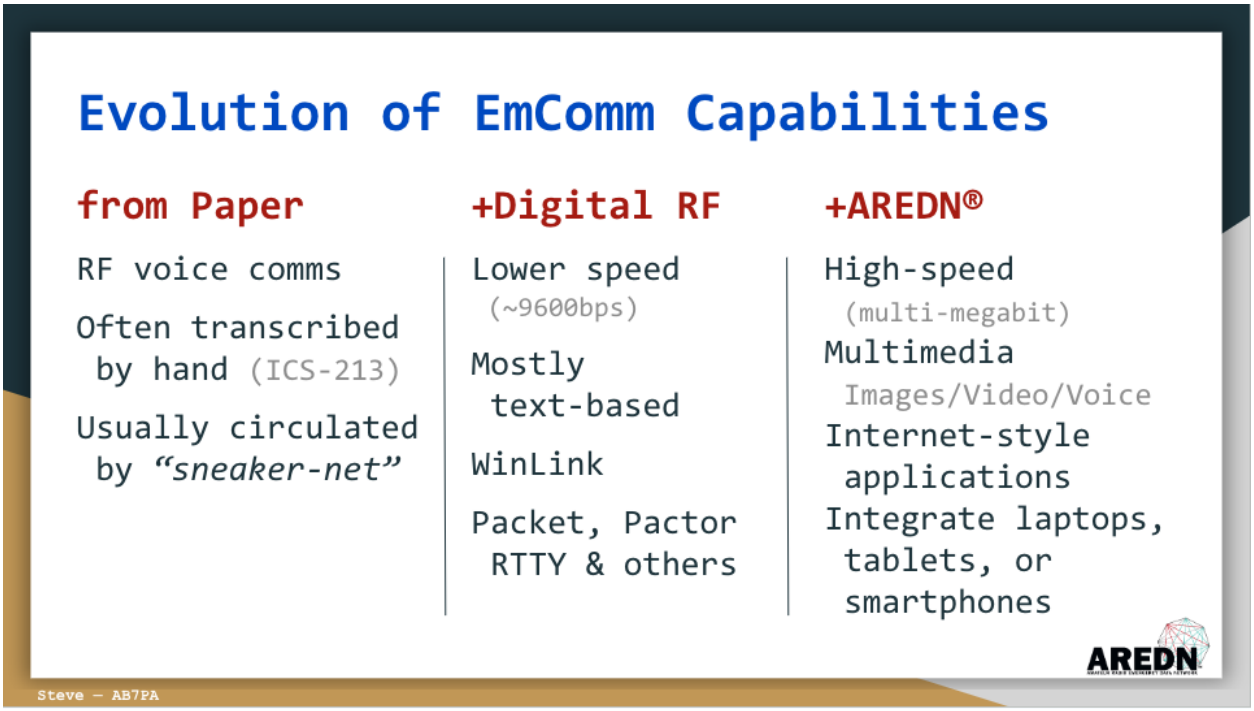
TWO

WHY AREDN®

AREDN® provides a way for amateur radio operators to create high-speed data networks for use in emergency and community service communication. At a high level, an Amateur Radio Emergency Data Network is simply another tool for your EmComm toolbox. As an amateur radio operator involved in emergency communication, you already have quite a few RF resources that you use on a regular basis. AREDN® is yet another tool that you might want to have available if it meets an important EmComm requirement, which we'll see in a moment.

Some seasoned operators may ask, "Why do I need another tool when the ones I already have are working just fine?" The simple answer is that you only need AREDN® when it serves a useful purpose or meets an important need for your served agency. As always, you should use the right tool for the job.

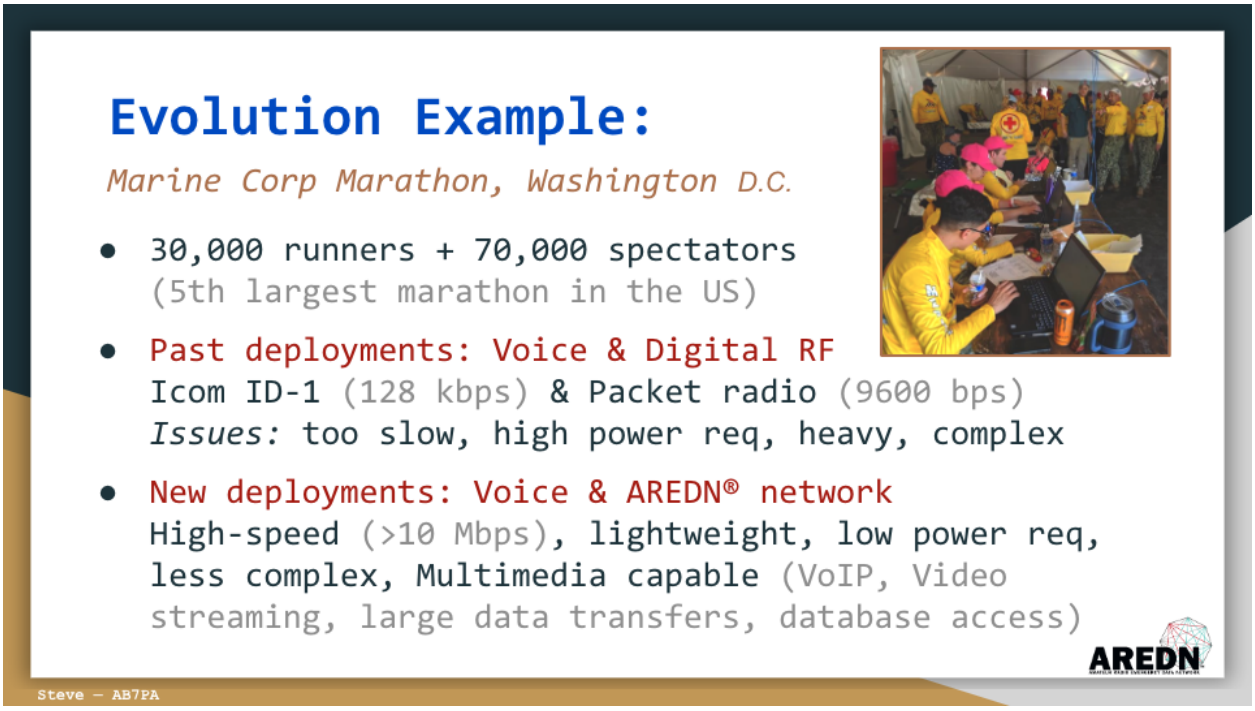
When might you want to use AREDN® mesh networking? It depends on what type of communication is required for your deployment. AREDN® is very useful if your served agency needs specific applications or services that require a computer network between sites. If high-speed digital communication is needed across an area, then AREDN® is a good solution. If the sites to be linked are located in areas where normal infrastructure has become unavailable, then AREDN® nodes can be used to create a portable off-grid data network. Also, if different resources are transient because they come and go at various locations, then an AREDN® node's ability to automatically join or form mesh networks might be a real benefit. It might be helpful to look at the evolution of EmComm capabilities used by amateur radio operators through the years.



Traditionally we have used RF voice communication on a variety of radio bands. A typical message-passing scenario involved giving the message to a radio operator who would write or type it onto a standard ICS-213 form. The message would then be relayed by RF voice comms to another operator who would write or type it on another ICS-213 form at the receiving end. The form would typically be hand-delivered to the recipient who would read and acknowledge the message. Any reply would then be handled through the same process from the receiving end back to the originator. This tried-and-true method has worked well, and it continues to work for handling much emergency and event traffic.

In recent years Digital RF communication was included in the EmComm toolkit, with the addition of things like Packet Radio and WinLink. These modes moved emergency message passing into the digital realm, and this minimized or eliminated some of the sources of error in the communication chain. Digital RF communication was mainly text-based and is relatively slow speed but very reliable.

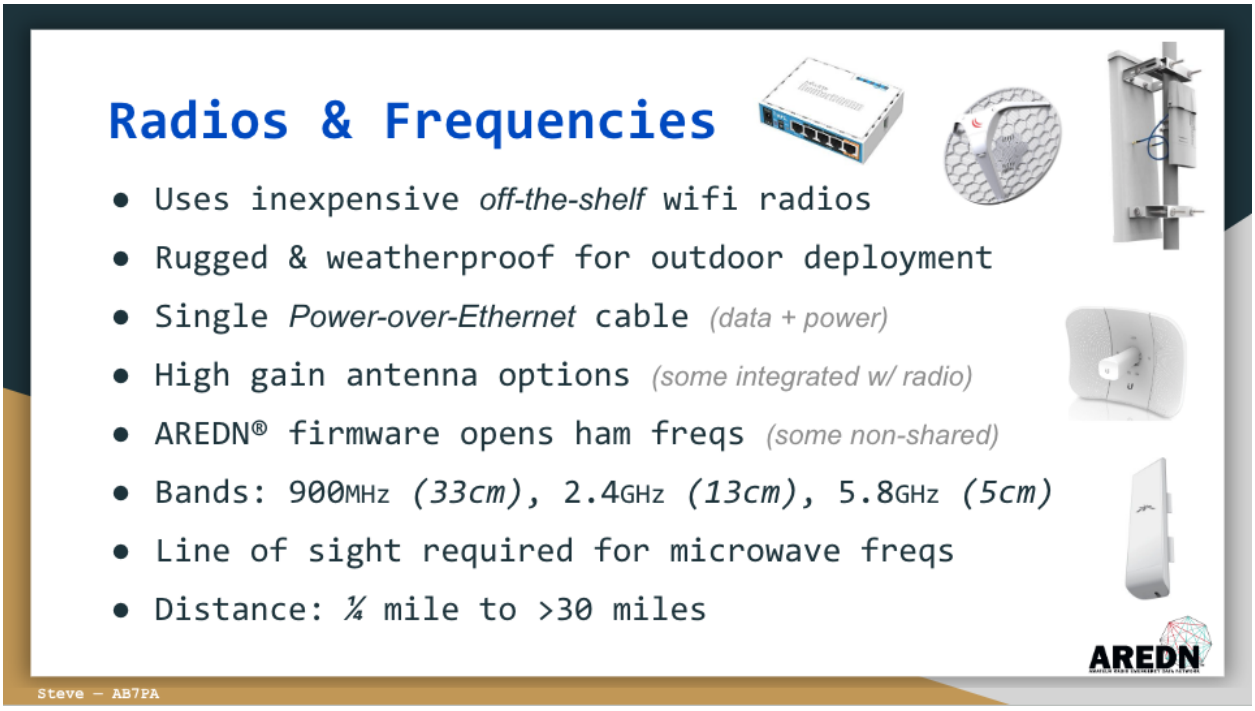
When AREDN® became available it added several features which the served agency staff were already familiar with in their normal operations. These include the ability to transfer digital messages at relatively high speeds (in the multi-megabit range), as well as the capability for multimedia communication such as Voice, Photos, and streaming Video. It gave them the ability to use Internet-style applications or programs, and to integrate their smartphones, tablets, and laptops into the EmComm network. Let's take a look at one example of how amateur public service communication has evolved over the years.



This photo was taken at a recent Marine Corp Marathon in the Washington, DC area. It's one of the largest marathons in the country, with around 100,000 participants and spectators. A group of amateur radio operators has been providing communication services for this event for many years.

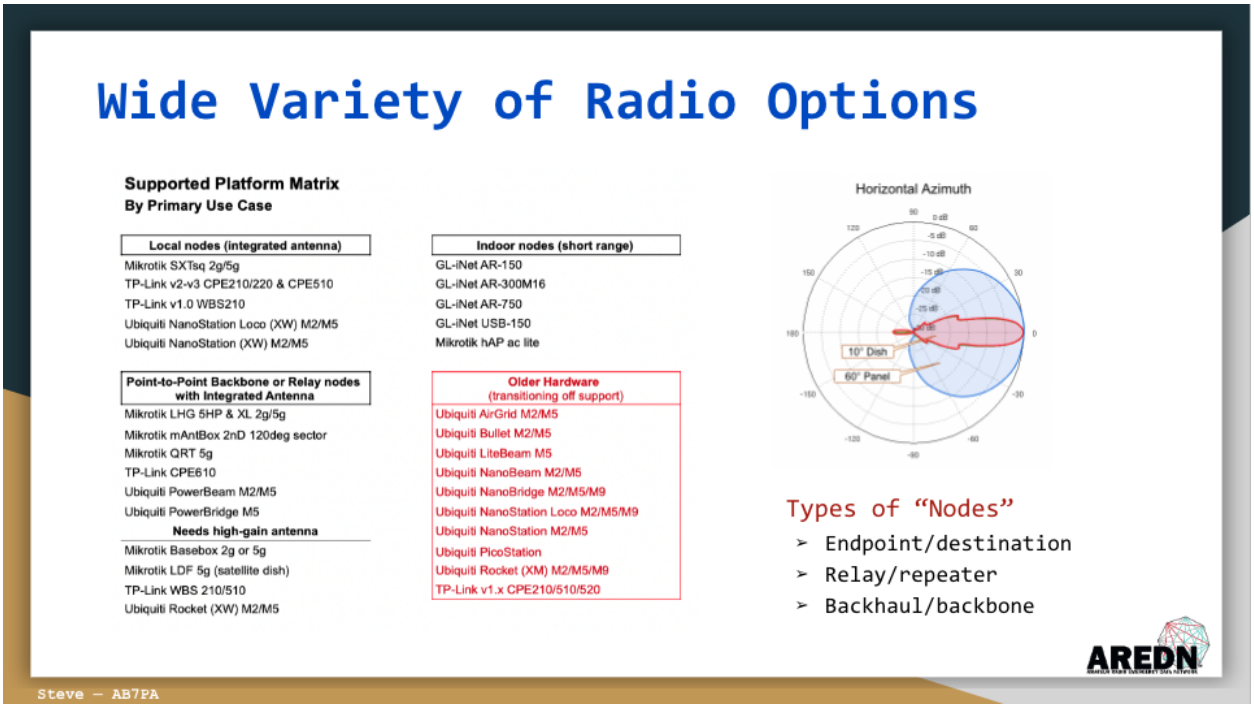
In the past they used mainly Voice & Digital RF modes. Typically they deployed Icom ID-1 data radios with speeds around 128 kbps, as well as packet radio with speeds around 9600 baud. Their After Action Reports identified several concerns, though. They indicated that data transfer rates were too slow. They also mentioned that the equipment was heavier and more complex to set up, as well as requiring higher capacity portable power sources.

For a couple of years they began experimenting with AREDN®, and recent deployments have been based around AREDN® networks. They indicated that they were able to achieve high data transfer speeds (in the range of 10 Mbps or more), using equipment that was lighter weight, less complex, and required much less power to operate. In addition, they were able to provide Voice over IP, Video Streaming, and multi-user network database access.



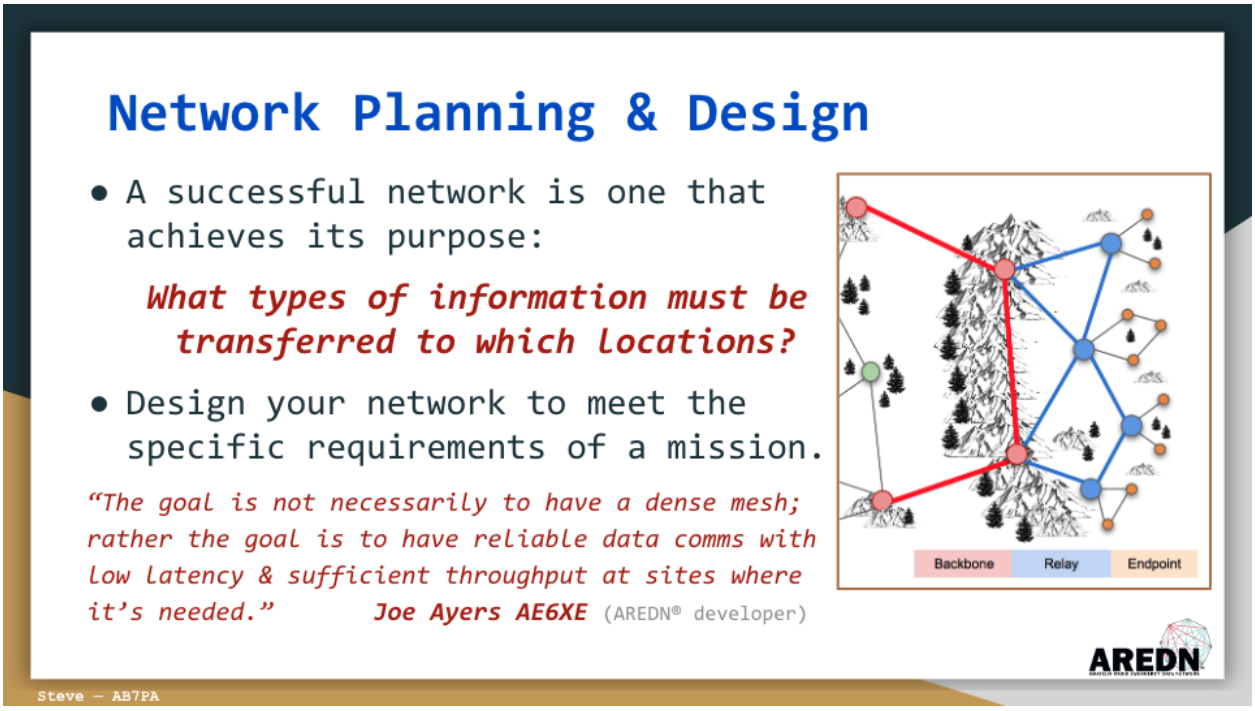
Devices that support AREDN® come in a wide variety of shapes and sizes because AREDN® firmware can be installed on many types of inexpensive “off the shelf” wifi radios. AREDN® allows us to repurpose commercially available radios as mesh network nodes, many of which can communicate on unshared frequencies set aside specifically for licensed amateurs. Most of these commercial radios are rugged and weather-proof for outdoor installations. They typically use Power over Ethernet (PoE) which makes them less complicated to deploy by having a single cable to the device. Many of them also have integrated high gain antennas.

The frequency ranges that are currently supported are the 900 MHz, 2.4 GHz, and 5.8 GHz bands. These microwave frequencies do require direct line of sight for reliable communication. Depending on the type of radios and antennas that are deployed, it’s possible to achieve network links anywhere from a few miles to well over 30 miles between sites.



There are many radio options for different use cases. Some nodes are small indoor-only devices that provide both Ethernet and standard WiFi coverage, as well as mesh RF coverage out to a limited distance. Other radios are intended to provide mesh RF coverage to a 90 or 120 degree sector for devices at longer distances. Still other radios are designed for narrow beam, high speed, point to point data transfer, typically between mountaintop or backbone locations.

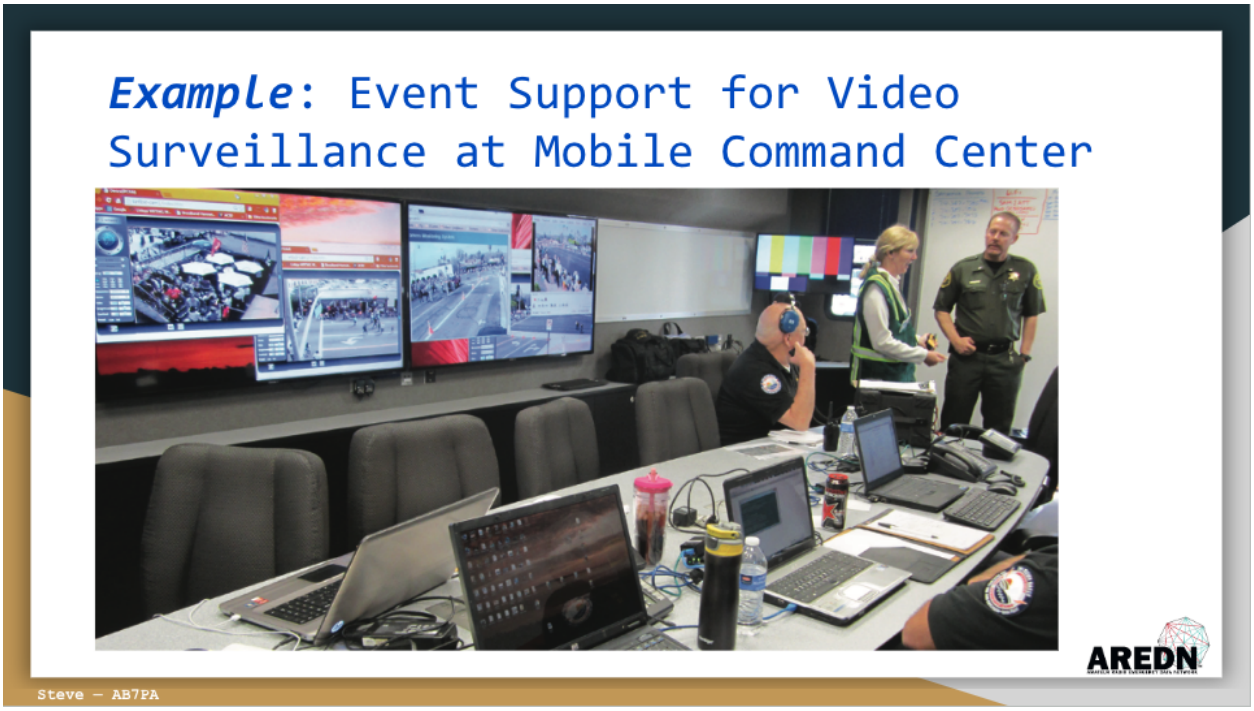
From this you can see that different types of radios serve different functions within the wider mesh network. You could categorize them as local endpoint nodes, or intermediate relay nodes, or high-speed backhaul nodes based on the purpose they serve for the network as a whole.



For our purposes in providing emergency or event communication as volunteers, we should focus on designing a network that is able to reliably transfer information to and from the locations where it is needed. A successful network is one that achieves its purpose, so design your networks to meet the specific requirements of a mission. We'll see some good examples of this in a moment, but we should keep this as our main goal for using AREDN® to provide EmComms.

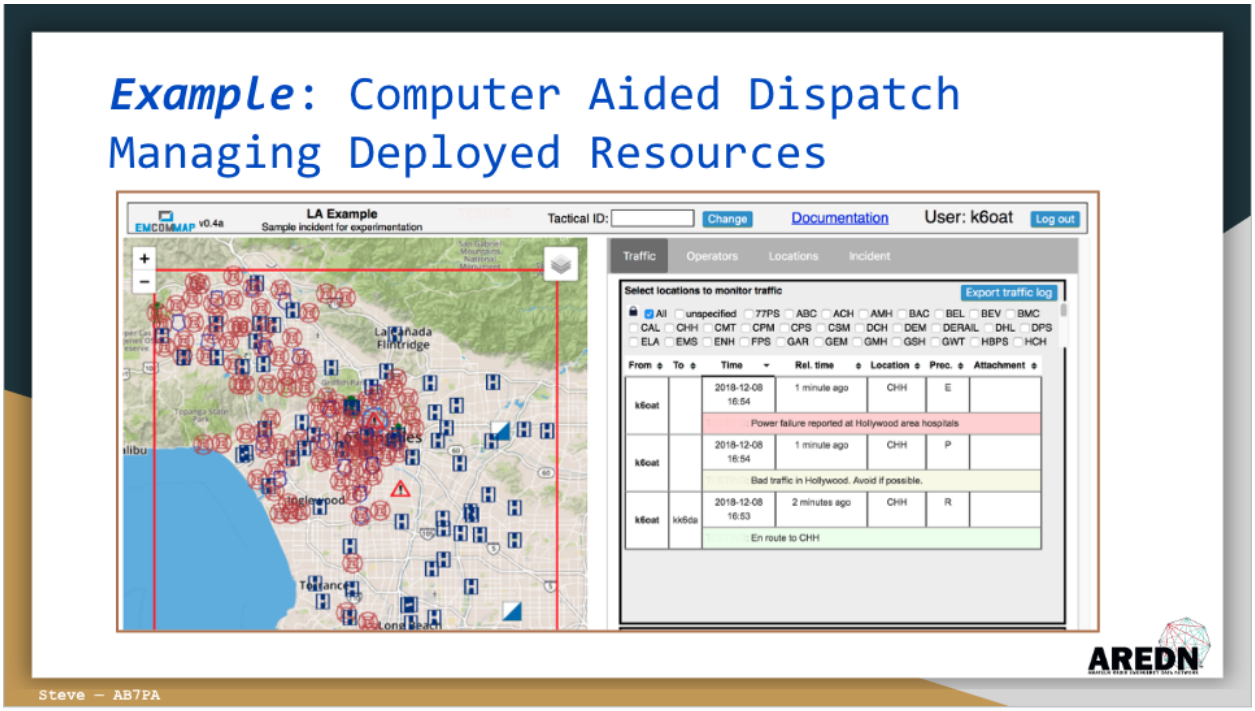
In simplest terms, when you deploy AREDN® devices you are providing a high-speed digital data network. Keep in mind that the network itself doesn't really accomplish your mission. The applications, programs, and services riding your network are the key to accomplishing the mission. Your group may or may not be responsible for providing those applications and services. But if you do provide a program or service, be sure that what you provide is simple and intuitive to use, both for other amateur operators as well as for served agency staff members. Let's look at some specific examples.

Whenever possible, deploy services that people are already familiar with. These days anyone can pick up a telephone and call someone's phone number. They're used to chatting with friends and co-workers using their computer keyboard. Almost everyone can read and send an email, often with large files or photos as attachments. People are used to pulling up a photo or image in their web browser or watching a streaming video from a web site. These are the types of services that would be a good fit for AREDN® networks. In this example, several stations were set up as part of an EmComms exercise. Participants were able to pick up a standard telephone to dial or answer phone calls between distant locations, all transmitted by RF using an AREDN® network.



In this example, an amateur radio group was given the mission to provide live video feeds across a specific area. AREDN® nodes with video cameras were deployed at key points along the route, and network connected computers displayed each video stream on different monitors in the Sheriff's mobile command post.

After this event someone from the served agency said, "This mesh camera system provided by RACES members was a valuable tool for our command staff. The parade was the safest in years. As we were taking the calls, we could see the activity occurring in real time. Incredibly, there was only one arrest for fighting, which just happened to take place in the camera's view."

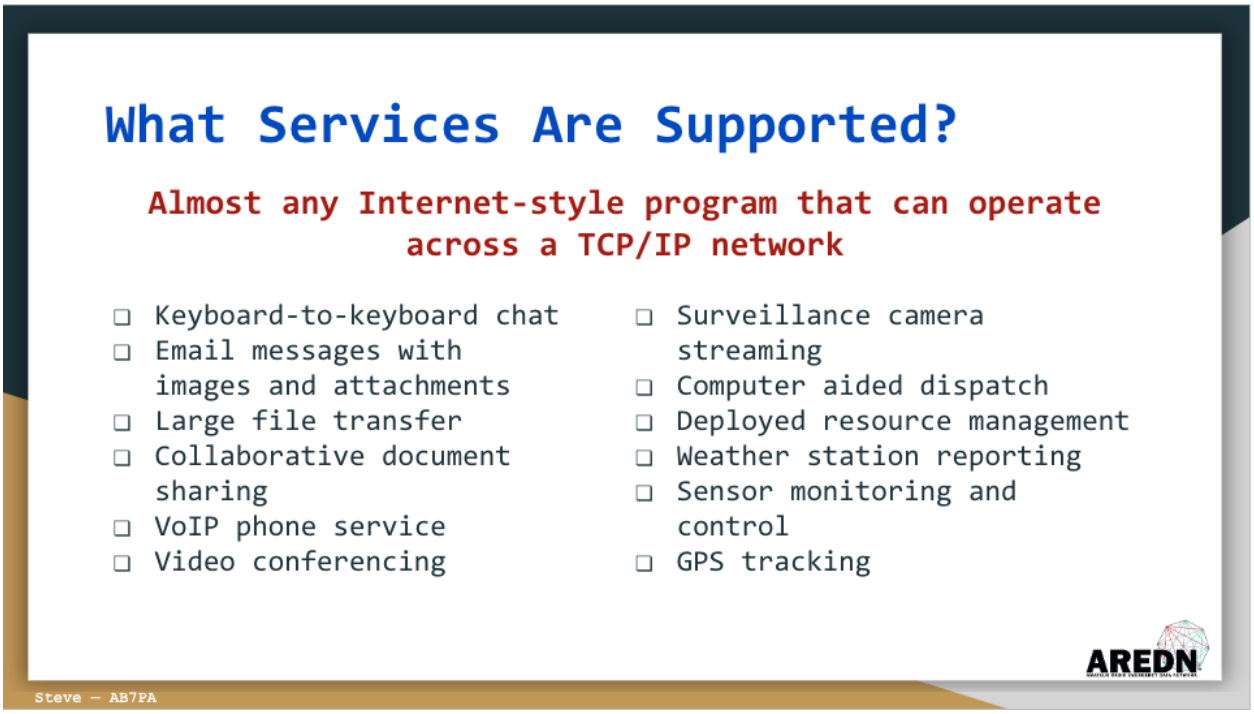


When a community-wide event or emergency occurs, one of the challenges is keeping track of deployed resources – whether they are people, or places, or equipment. In this example, an AREDN® network is being used to track resources and display messages that are sent between sites. The map on the left is a great visualization tool, and the main goal of this application is to increase the team's situational awareness. The specific software running on this mesh network was developed by Dan K6OAT for the Los Angeles ARES team. People at each location are able to see what is going on around them from their mesh-connected computer.

In southern California some of the mountaintop AREDN® backbone sites were deployed with video surveillance cameras on the towers. In this example, one of these mountaintop cameras captured and recorded this image. It was the first view of the 2017 Thomas Fire. This recording was requested by the fire management authorities to be included in their after action reports.

The inset on the right is an image of the flight paths of tanker aircraft traversing the region. Flight data was captured using an ADS-B receiver and displayed from a Raspberry Pi computer on the AREDN® network.

This example illustrates using small AREDN® nodes connected to agency laptops. Each computer then has access to the AREDN® network and has the ability to communicate with other network resources. This would provide local communication across a field or parking lot as shown here, but the laptops could also link to an intermediate AREDN® node on top of a mast in the center of the area. From there the data could be transferred across longer distances to sites that are coordinating the event or exercise.



Almost any Internet-style program that operates across a standard TCP/IP network can be deployed using AREDN® devices. This includes all of the examples shown in this list. Just remember that the services deployed should align with the specific mission or purpose for the network you are creating. Just because you can add nodes or services to a network, doesn't mean you should add them. Each new item added to a network will use part of the limited processing and bandwidth resources that are available. Make sure your network is successful by deploying exactly what is needed in order to accomplish your mission.

Probably the best single place to go for additional information is the AREDN® website at www.arednmesh.org. There you will find information about the types of radios that are supported, as well as all of the AREDN® software available for download.

There is also a wealth of information on choosing devices and planning AREDN® networks for EmComms. The Forum provides a way to engage with a very active worldwide community of fellow hams who are working with the same hardware and software that you are. They are eager to help answer questions, as well as testing various devices and network configurations.

Regional and local AREDN® mesh groups can also be contacted through the Forum. You can also access the extensive set of documentation that is available online, including detailed sections on installing and configuring radios, planning and modeling network links, providing different kinds of services for your network, and a variety of other topics.

CHAPTER

THREE

SELECTING RADIO HARDWARE

The amateur radio community has recognized the benefits of using inexpensive commercial WISP (Wireless Internet Service Provider) radios to create AREDN® networks. Each of these devices comes with the vendor's firmware preinstalled, but by following documented procedures this firmware can be replaced with an AREDN® firmware image.

Several open source software projects have been adapted and enhanced to create the AREDN® firmware, including OpenWRT and Babel. The AREDN® team builds specific firmware images tailored to each type of radio, and the current list of supported devices is found on the AREDN® website. For a complete list of all supported hardware, including both *Stable Release* and *Nightly Build* firmware, refer to the Supported Devices list. If at all possible try to avoid using devices listed under the *Sunset* heading, since those older devices are being retired.

When selecting a device for your AREDN® hardware there are several things to consider in your decision.

- Radios should be purchased for the specific frequency band on which they will operate. Currently AREDN® supports devices which operate in several bands. Check the frequency and channel chart on the AREDN® website for the latest information.
- Radios can be purchased separately from the antenna, so it is possible to have more than one antenna option for a radio in order to optimize AREDN® nodes for varying deployment conditions.
- Costs of devices range from $25 to several hundred dollars for a complete node/antenna system, so there are many options even for the budget-conscious operator.
- Some older or lower cost devices have a limited amount of onboard memory. Consider purchasing a device with more memory over one with less memory.
- Check the maximum power output of the device, since some devices have lower power capabilities.

One of the best sources of detailed hardware information is a manufacturer's datasheet, usually available for download from the manufacturer's website. Currently AREDN® supports dozens of device models from manufacturers including GL-iNet, Mikrotik, MorseMicro, TP-LINK, and Ubiquiti Networks.

If you are just getting started with AREDN® you can easily begin with one of the low-cost devices that comes with an integrated antenna and a PoE (Power over Ethernet) unit. If you are expanding your AREDN® network with more sophisticated equipment, you may choose a standalone radio attached to a high-gain antenna.

Note: See the **Network Design Guide** for more information about constructing robust mesh networks.

CHAPTER

FOUR

DOWNLOADING AREDN® FIRMWARE

4.1 Types of Firmware

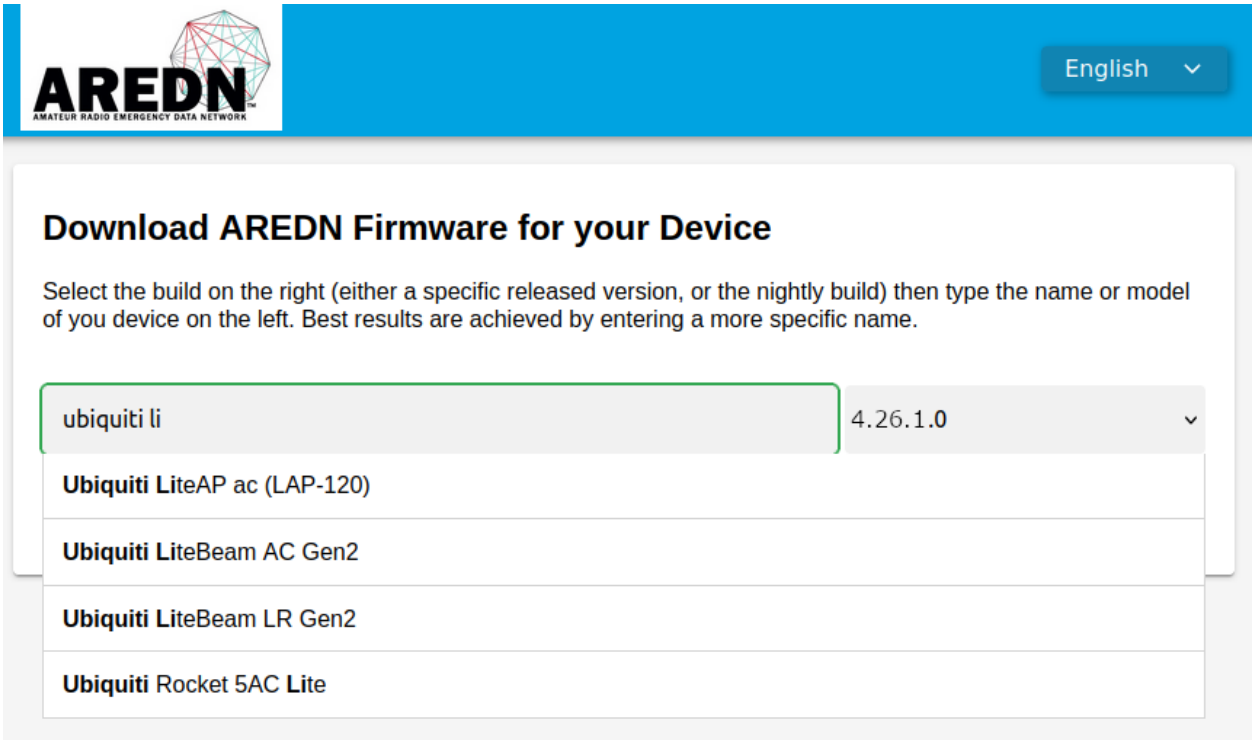
Stable Release firmware has been tested and shown to work on the devices that were supported at the time of the release. This firmware is considered to be stable and suitable for production devices deployed in the field. Stable Release firmware is identified by numbers such as `4.26.1.0`. In this example `26.1` indicates the year (2026) and month (Jan) of the release.

Nightly Build firmware contains the latest bug fixes, features, and support for new devices. It allows the wider mesh community to test new code before it is included in a Stable Release. The Nightly Build is considered more experimental or cutting-edge and may not be suitable for production nodes. However, it might make sense to install the Nightly Build if you are having a specific issue that has been addressed in newly developed code or if you are loading AREDN® firmware onto a device that is newly supported. The Nightly Build filename shows the build date and the software commit identifier for that specific firmware build.

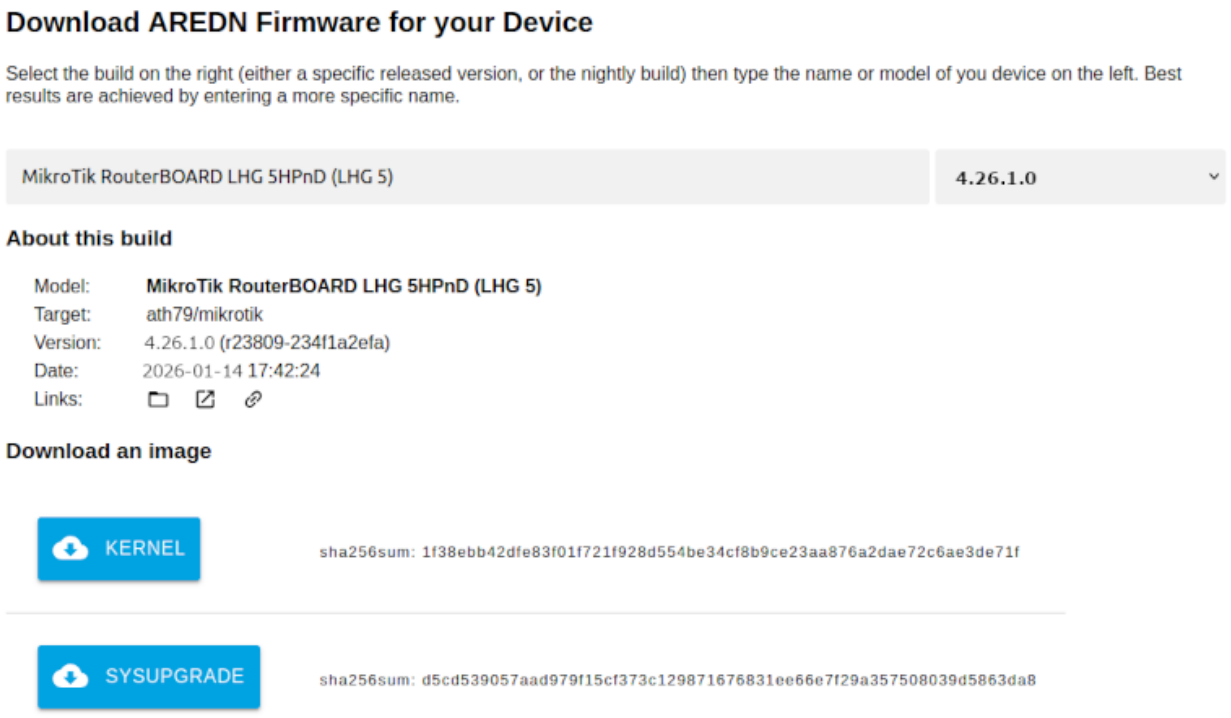
Attention: Be aware that when a new nightly build appears, the older build automatically becomes obsolete. If you want to install add-on packages for nodes running a nightly build, understand that specific packages are not available for an *older* build if a *newer* build has superseded it. You will need to upgrade to the current nightly build before installing packages.

4.2 Choosing Firmware to Download

The first step is to choose the AREDN® firmware image for your specific hardware. You can find the available firmware images for your device by using the AREDN® Firmware Selector (AFS).



Enter the first few characters of the hardware manufacturer in the *Model* search field (case insensitive), then click the firmware image dropdown on the right to choose the firmware release that you want to download. Next, find your device model in the search results list and click the row for your hardware. As shown below, this display will have links for all of the firmware images that are available to download for your device.



There are usually two types of firmware images shown for each device: one for the first-time replacement of the manufacturer's firmware, and the other for upgrades of nodes that are already running AREDN® firmware.

TP-LINK or Ubiquiti
: If you are loading firmware on TP-LINK or Ubuquiti devices for the first time you must download the *FACTORY* firmware. Otherwise download the *SYSUPGRADE* firmware image.

Mikrotik
: If you are loading firmware on Mikrotik devices for the first time you must download **both** the *KERNEL* and *SYSUPGRADE* images. Otherwise download only the *SYSUPGRADE* firmware image.

  As noted in the Mikrotik install instructions, if you determine that your device is running RouterOS `v7.x` you can try to install the *SYSUPGRADE-V7* image or you can follow the procedure to downgrade RouterOS and then install the regular *SYSUPGRADE* image.

GL.iNET, MorseMicro, Cudy, and OpenWRT One
: These devices require only the *SYSUPGRADE* image for both first-time installs and firmware upgrades.

ZyXel
: ZyXel devices always take the *FACTORY* image.

Click the appropriate button to download the image file to your local computer. Make a note of the download location on your computer, since you will use the downloaded image(s) to install the AREDN® firmware on your device.

The latest AREDN® firmware contains features which are inherited from the newest OpenWRT upstream releases. The OpenWRT Release Notes describe these new features.

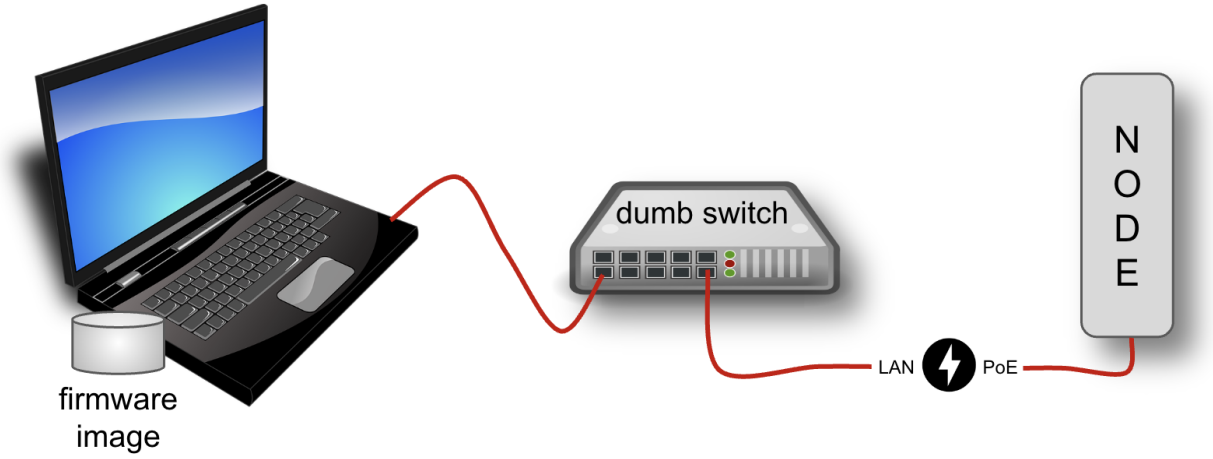
You should select the latest recommended target image based on the type of hardware on which it will be installed. Refer to the latest Supported Devices in order to ensure you have the correct firmware image for your specific device.

CHAPTER

FIVE

INSTALLING AREDN® FIRMWARE

The diagram below shows your computer with the downloaded firmware image connected to the node using Ethernet cables in order to install the AREDN® image. It is highly recommended that you connect the computer and node through a simple (dumb) Ethernet switch so that the switch can maintain the computer's network link even when the node is rebooting. Do *not* use a network router for this purpose – only a dumb switch. This is not for the sake of the radio, but it allows your computer to maintain its Ethernet interface link even when the node reboots.



In the *Firmware Tips* section of the **How-To Guide** you will find assistance if you experience an issue uploading firmware to your device. The **How-To Guide** also contains a *Virtual Machine Installs* section for help installing x86_64 firmware images on a VM for a virtualized node.

5.1 Preparing your computer

Setting a Static IP Address on your Computer

For all of the device models discussed below you will be asked to set a static IP address on your computer as part of the install process. Various computer operating systems have different ways of accomplishing this, so you should check your computer manuals, publications, and online resources to walk you through the steps for your specific computer.

As mentioned above, AREDN® recommends that you connect your computer to the node through an intermediate network switch. This allows your computer to activate its Ethernet interface with the static IP address even when the node is not powered on. Since node hardware needs to be powered on/off or rebooted during the install process, the network switch will keep your computer's network interface active on its static IP address.

If you choose not to use an intermediate network switch, then you will be responsible for making sure your computer maintains an active interface with the static IP address. You may need to power on the node temporarily in order for your computer to bring up its interface, but then immediately power off the node in order to follow the installation instructions for your model. Having an intermediate network switch eliminates these steps.

Depending on your computer operating system you may not have various command line tools available on your computer. The required tools are available by default on Linux, MacOS, and Windows 11, but you may need to enable specific features or install appropriate programs as noted below.

For Ubiquiti 802.11n Installs

Your computer should have TFTP *client* software available. If you have a Windows computer, use a web search engine to find information for your specific operating system (for example search "tftp client for windows"). There is a wealth of information available online for configuring your Windows computer with a TFTP client program.

For Ubiquiti 802.11ac Installs

Your computer should have ssh and scp software available. *Ssh* and *scp* are available by default on Linux, MacOS, and Windows 11. On Windows computers you may also use programs such as PuTTY and WinSCP to connect to your device.

For Mikrotik and TP-LINK Installs

These devices are programmed to download a boot image from an external source. Your computer will provide the Preboot eXecution Environment (PXE) which will give the node an IP address via DHCP as well as providing the firmware image via TFTP.

- If you have a Windows computer you must install and configure a PXE *server*. The examples below use *Tiny PXE* which can be downloaded from erwan.labalec.fr. There may be other alternative Windows programs that accomplish the same goal, such as ERPXE or Serva. For TP-LINK devices you may be able to run a simple TFTP server such as Tftpd64 as explained in the TP-LINK section below.
- If you have a Linux or MacOS computer, your "Preboot eXecution Environment (PXE)" will be provided by the native `dnsmasq` program, as described in the Linux procedures

below.

5.2 Ubiquiti 802.11n first install process

These devices have a built-in TFTP *server* to which you can upload the AREDN® *factory* image. Your computer must have TFTP *client* software available. For more information, see the **Preparing Your Computer** section above.

Different TFTP client programs may have different command line options or flags that must be used, so be sure to study the command syntax for your TFTP client software. The example shown below may not include the specific options required by your client program.

Download the appropriate *factory* file for your device by following the instructions in the **Downloading AREDN® Firmware** section of this documentation.

1. Set your computer's Ethernet network adapter to a static IP address that is a member of the correct subnet for your device. Check the documentation for your specific hardware to determine the correct network number. As in the example below, most Ubiquiti devices have a default IP address of 192.168.1.20, so you can give your computer a static IP on the 192.168.1.x network with a netmask of 255.255.255.0. For example, set your Ethernet adapter to a static IP address of 192.168.1.10. You can choose any number for the fourth octet, as long as it is not the same as the IP address of the node. Of course you must also avoid using 192.168.1.0 and 192.168.1.255, which are reserved addresses that identify the network itself and the broadcast address for that network. Other devices may have different default IP addresses or subnets, so select a static IP for your computer which puts it on the same subnet but does not conflict with the default IP of the device.
2. Connect an Ethernet cable from your computer to the dumb switch, and another cable from the LAN port of the PoE adapter to the switch.
3. Put the Ubiquiti device into TFTP mode by holding the reset button while plugging your node's Ethernet cable into the *POE* port on the PoE adapter. Continue holding the device's reset button for approximately 30 to 45 seconds until you see the LEDs on the node alternating in a 1-3, 2-4, 1-3, 2-4 pattern, then release the reset button.
4. Open a command window on your computer and execute a file transfer command to send the AREDN® firmware to your device. Target the default IP address of your Ubiquiti node, such as 192.168.1.20 (or 192.168.1.1 for AirRouters). The TFTP client should indicate that data is being transferred and eventually completes. The following is one example of TFTP commands that transfer the firmware image to a node:

```
[Linux/Mac]
> tftp 192.168.1.20
> bin                 [Transfer in "binary" mode]
> trace on            [Show the transfer in progress]
```

(continues on next page)

(continued from previous page)

```
> put <full path to the firmware file>
  [For example, put /tmp/aredn-<release>-factory.bin]
----------------------------------
[Windows with command on a single line]
> tftp.exe -i 192.168.1.20 put C:\temp\aredn-<release>-factory.bin
```

5. The node will now automatically reboot with the new AREDN® firmware image.

5.3 Ubiquiti 802.11ac first install process

Contributor: Tim Wilkinson KN6PLV

Prerequisites

The installing computer must be capable of connecting to the command line of the target device. This will require that the computer support both the *ssh* and *scp* protocols. *SSH* and *scp* are native to both Linux and MacOS. The OpenSSH package (which contains both commands) can be enabled on Windows computers. For more information, see the **Preparing Your Computer** section above.

Step 1: Preparing the device

Before you install AREDN® firmware on a Ubiquiti 802.11ac device, you must first make sure it is running a specific version of the standard Ubiquiti AirOS software. This procedure will not work if the device is running any other version. Fortunately you can upgrade or downgrade the standard Ubiquiti software.

As described in the first paragraphs of this document, it is best to connect your computer to the device using a simple Ethernet switch so that your computer's network interface remains unaffected by reboots on the radio. The IP address for a new Ubiquiti device is 192.168.1.20. Set the IP address of your computer to 192.168.1.10 and, when the device is powered up, enter 192.168.1.20 in a web browser. For a brand new device you'll be asked to select your country and agree to the EULA. Then click *Continue*. Next you will be prompted to create a user account and password on the radio. You can enter the username `admin` and the password `admin!23` (for example) and then click *Save*. Make a note of this username and password because you will use it in the following steps.

You should now see the main Dashboard view in AirOS. On the left, click the *Gear* icon. This will take you to the System page. At the top of this page you will find the radio's current firmware version. For example, it might read `FIRMWARE VERSION XC.V8.7.1`. If the firmware version shows either **XC.V8.7.0**, **WA.V8.7.0** or **2WA.V8.7.0** then you have the correct AirOS software and can move on to **Step 2**.

But if you see any version other than 8.7.0 you must upload new firmware to the device. You will need to download the correct firmware to your installing computer. The firmware can be found here:

- WA: https://dl.ubnt.com/firmwares/XC-fw/v8.7.0/WA.v8.7.0.42152.200203.1256.bin
- XC: https://dl.ubnt.com/firmwares/XC-fw/v8.7.0/XC.v8.7.0.42152.200203.1256.bin
- 2WA: https://dl.ubnt.com/firmwares/XC-fw/v8.7.0/2WA.v8.7.0.42152.200203.1256.bin

Select the firmware appropriate for your device. If the radio's current firmware starts with *WA* download that version. If it starts *XC* download that version. If it starts *2WA* download that version.

On the top right of the System page you will see "UPLOAD FIRMWARE" and UPLOAD in blue. Clicking the blue UPLOAD text will open a dialog and let you select the **8.7.0** firmware you downloaded to your computer. Now that firmware will be uploaded to the device. Once completed a dialog in the top right will be displayed allowing you to either UPDATE or DISCARD the newly uploaded firmware. Click *UPDATE*. The upgrade process will now start. Do **not** unplug the device until this step is completed.

Once the upgrade has been completed, the device will return you to the login page. Log in using the username and password you created earlier (`admin` / `admin!23`). Once again you will see the System page and if everything has been successful, the firmware version will now read either WA.V8.7.0, XC.V8.7.0 or 2WA.V8.7.0 and you can move to **Step 2**.

> **Attention:** The upgrade can fail on newer hardware which requires **8.7.4** firmware. This problem has been observed and tested on newer LiteBeam AC, PowerBeam AC and NanoBeam AC devices. For these devices, follow the same firmware downgrade procedure but use the following firmware instead:
>
> - WA: https://dl.ubnt.com/firmwares/XC-fw/v8.7.4/WA.v8.7.4.45112.210415.1103.bin
>
> The rest of the process remains unchanged, so once the downgrade is successful you can move to **Step 2**.

Step 2: Copy the AREDN® firmware to the device

Before you can install AREDN® firmware on the device, you first need to put the AREDN® image in the device's `/tmp` directory. Note that each 802.11ac model will have a *different* AREDN® image name, as opposed to past releases where one AREDN® image supported multiple models. Be sure to download the correct firmware image from the AREDN® download site. On your computer, open a terminal session ("CMD" in windows). Copy the firmware to the device using the scp command with the username and password you created in **Step 1**. The example command below shows the placeholder `<aredn-image-factory.bin>` for the firmware filename, but be sure to replace this with the actual filename of the firmware you are installing.

```
scp <aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

If you see the error "Unable to negotiate" it means that the SCP program you are using on your computer does not support the default security key type being used on the device. You

should refer to the documentation for that SCP program to resolve the issue. You can try the following:

```
scp -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-rsa
↪<aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

If you see an error "sftp-server: not found" you can try the following:

```
scp -O -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-
↪rsa <aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

If you see an error "Remote host identification has changed" you can try the following:

```
scp -O -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-
↪rsa -oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no
↪<aredn-image-factory.bin> admin@192.168.1.20:/tmp/factory.bin
```

Once this is successful, the AREDN® firmware will be in `/tmp` on the device waiting to be installed.

Step3: Install the firmware

The installation procedure requires you to **ssh** to the command line of the device. On your computer, open a terminal session ("CMD" in windows). Type or copy/paste the following command:

```
ssh admin@192.168.1.20
```

If you see the error "Unable to negotiate" please try the following:

```
ssh -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-rsa␣
↪admin@192.168.1.20
```

If you see an error "Remote host identification has changed" you can try the following:

```
ssh -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedAlgorithms=+ssh-rsa␣
↪-oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no␣
↪admin@192.168.1.20
```

You will be asked for the password created in **Step 1** (for example, admin!23) and once entered you will be logged into the device and shown the shell prompt.

To install the AREDN® firmware you first need to create a program to do this. Ubiquiti devices expect signed firmware but AREDN® is not signed, so we need to bypass the checking process. To do this type or copy/paste the following two commands:

```
hexdump -Cv /bin/ubntbox | sed 's/14 40 fe 27/00 00 00 00/g' |␣
↪hexdump -R > /tmp/fwupdate.real
```

(continues on next page)

(continued from previous page)

```
chmod +x /tmp/fwupdate.real
```

These commands take the standard Ubiquiti program used for flashing new firmware and change a few bytes to create our own version with the signature checking code disabled. The first command can take a little while to complete but when successful will return you to the shell prompt.

Finally flash the AREDN® firmware by typing:

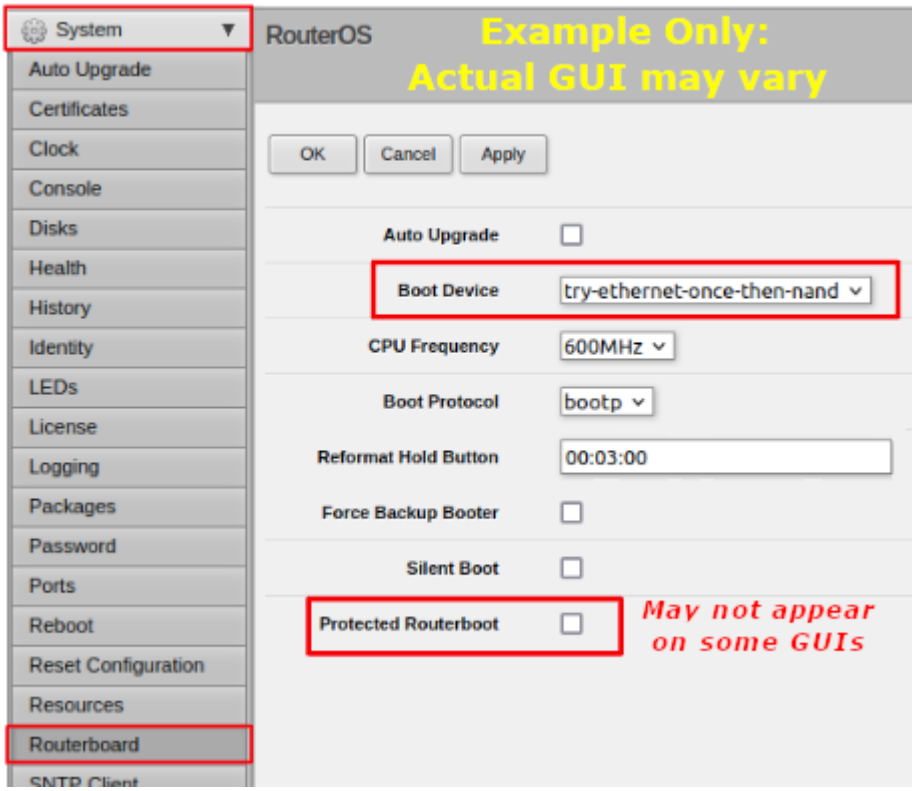
```
/tmp/fwupdate.real -m /tmp/factory.bin
```

Do **not** unplug the device until the flashing process is complete and the device has rebooted. The device will install the AREDN® image, boot into it, and end up on IP address 192.168.1.1 as a normal AREDN® device. If you cannot connect to the device on its new IP address after five minutes, power cycle the device and try connecting to 192.168.1.1 again. You can then configure the device by following the steps in the **Basic Radio Setup** section of the documentation.

5.4 Mikrotik first install process

These devices require a **two-part install** process: First, boot the correct Mikrotik *initramfs-kernel* file, and then use that temporary AREDN® environment to complete the installation of the appropriate *sysupgrade* file.

Mikrotik devices have a built-in PXE *client* which allows them to download a boot image from an external source. See the **Preparing Your Computer** section above for an explanation. The Windows example below uses *Tiny PXE*, while the Linux example uses the native `dnsmasq` program.



For Mikrotik devices you will use what is called *Etherboot* mode, and there are several ways to put your device into *Etherboot* mode (depending on the version of the manufacturer's firmware it is running). The easiest way is to use the device's reset button as described in the procedure below. If for some reason this does not work, then you can try logging into the Mikrotik RouterOS and setting *System* > *Routerboard* > *Settings* > *Boot Device* to `try-ethernet-once-then-nand` (either through the RouterOS web interface or via command line). Next time the device boots it will try *Etherboot* once before defaulting back to regular boot mode.

If your Mikrotik device has "Protected Routerboot" enabled, then you will need to disable it before proceeding. Use the manufacturer's instructions to connect to your device and display the RouterOS web interface or command line. Navigate to *System* > *Routerboard* > *Settings* > *Boot Device* to uncheck or deselect `Protected Routerboot`. Click the *Apply* button, then you should be able to power down the device and continue with the steps in the AREDN® firmware install checklist.

Tip: There may be cases when your Mikrotik device boots the AREDN® *kernel* file but its RouterOS version does not allow the *sysupgrade* file to be installed. You can read the instructions on this page (OpenWRT - Procedures for RouterOS) to determine which version of Mikrotik RouterOS your device has. If it is running version `7.x` then you can try installing the AREDN® *sysupgrade v7* firmware file. Or you can downgrade Mikrotik RouterOS prior to flashing the regular AREDN® *sysupgrade* file. Earlier versions of RouterOS and their NetInstall utilities can be found on the Mikrotik website. Download an ARM version (`routeros-arm`) for devices that use the *ipq40xx* AREDN® firmware, or download a MIPSBE version (`routeros-mipsbe`) for other Mikrotik devices. Typically you can install a RouterOS version that is equal to or newer than the

RouterOS version shown in the *Factory Firmware* field on the Mikrotik web display.

5.4.1 Mikrotik Install preparation

- Download *both* of the appropriate Mikrotik *kernel* and *sysupgrade* files from the AREDN® website. Rename the *initramfs-kernel* file to `rb.elf` and keep the *sysupgrade* **bin** file available for later.
- Set your computer's Ethernet network adapter to a static IP address on the subnet you will be using for the new device. This can be any network number of your choice, but it is recommended that you use the 192.168.1.x subnet. Using the 192.168.1.x network on your server will avoid having to change IP addresses on your computer during the install process. AREDN® firmware uses the 192.168.1.x network once it is loaded, so using it all the way through the process will simplify things for you. For example, you can give your computer a static IP such as 192.168.1.10 with a netmask of 255.255.255.0. You can choose any number for the fourth octet, as long as it is *not* within the range of DHCP addresses you will be providing as shown below.
- Connect an Ethernet cable from your computer to the network switch as described at the top of this document, then connect another cable from the LAN port of the PoE adapter to the switch. Finally connect an Ethernet cable from the *POE* port to the node, but leave the device powered off for now. If you are flashing a device which uses a separate power adapter (such as a *Mikrotik hAP ac* family device), connect the last Ethernet cable from the switch to the device's WAN port [1].

5.4.2 Boot the *kernel* image

Linux Procedure

If you are using a Linux or MacOS computer, use the following steps.

1. Create a directory on your computer called `/tftp` and copy the `rb.elf` file there.
2. Determine your computer's Ethernet *interface name* with `ifconfig`. It will be the interface you set to 192.168.1.10 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below. Use a `dhcp-range` of IP addresses that are also on the same subnet as the computer: for example 192.168.1.100,192.168.1.200 as shown below.
3. Open a terminal window to execute the following dnsmasq command with escalated privileges:

```
> sudo dnsmasq -i eth0 -u joe --log-dhcp --bootp-dynamic --dhcp-
↪range=192.168.1.100,192.168.1.200 -d -p0 -K --dhcp-boot=rb.elf --
↪enable-tftp --tftp-root=/tftp/
```

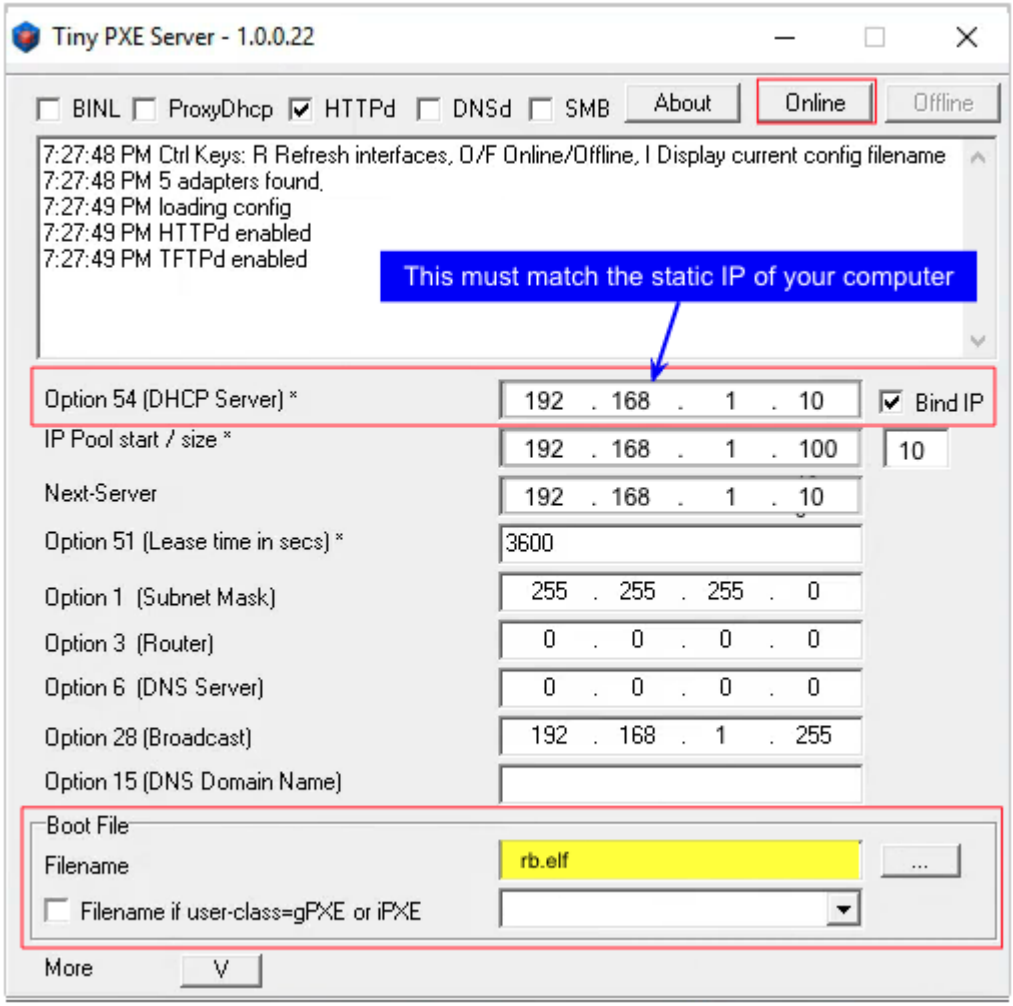
4. With the unit powered off, press and hold the reset button on the radio while powering on the device. Continue to hold the reset button until you see output information from the computer window where you ran the dnsmasq command, which should happen after 20-30 seconds. Release the reset button when you see the "sent" message, which indicates success, and you can now <ctrl>-C or end dnsmasq.
5. The node will now automatically reboot with the temporary AREDN® Administration image.

Windows Procedure

If you are using a Windows computer, use the following steps.

Configure the PXE Server on your Windows computer. The example below uses *Tiny PXE*. For more information, see the **Preparing Your Computer** section above.

1. Navigate to the folder where you extracted the *Tiny PXE* software and edit the `config.ini` file. Directly under the `[dhcp]` tag, add the following line: `rfc951=1` then save and close the file.
2. Copy the `rb.elf` file into the `files` folder under the *Tiny PXE* server directory location.
3. Start the *Tiny PXE* server exe and select your computer's Ethernet IP address from the dropdown list called `Option 54 [DHCP Server]`, making sure to check the `Bind IP` checkbox. Under the "Boot File" section, enter `rb.elf` into the the *Filename* field, and uncheck the checkbox for "Filename if user-class = gPXE or iPXE". Click the *Online* button at the top of the *Tiny PXE* window.



4. With the unit powered off, press and hold the reset button on the node while powering on the device. Continue holding the reset button until you see `TFTPd: DoReadFile: rb.elf` in the *Tiny PXE* log window.

5. Release the node's reset button and wait for the image to be transferred to the device. You are finished using *Tiny PXE* when the firmware image has been read by the node, so you can click the *Offline* button in *Tiny PXE*.

6. The node will now automatically reboot with the temporary AREDN® Administration image.

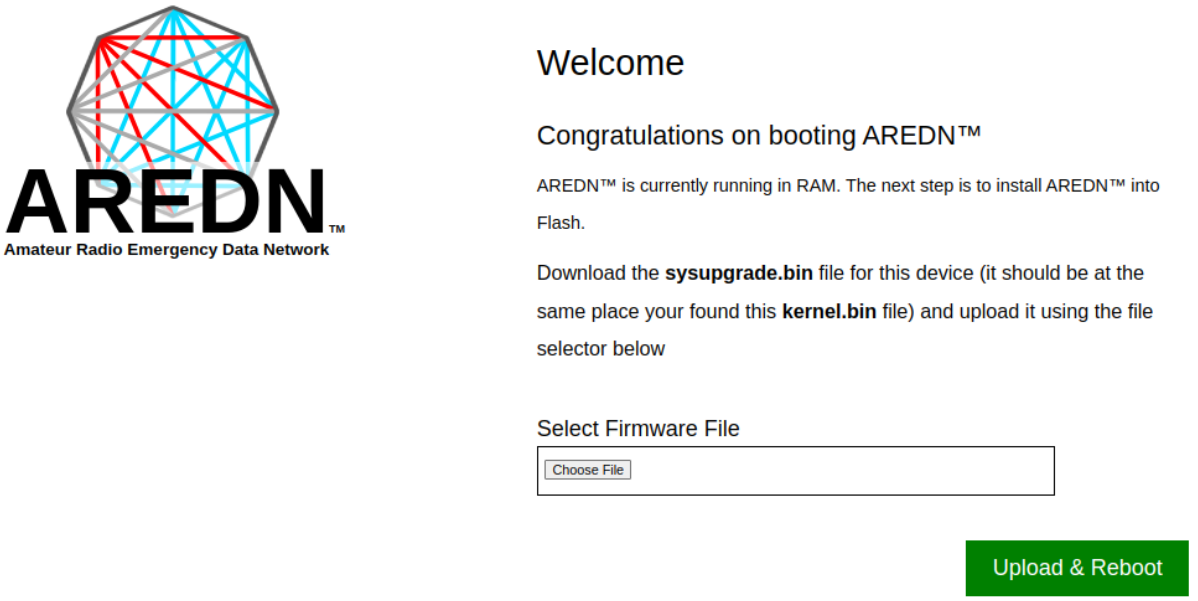
5.4.3 Install the *sysupgrade* firmware image

1. After booting the **kernel** image the node will have a default IP address of 192.168.1.1. Your computer should already have a static IP address on this subnet, but if not then give your computer an IP address on this subnet.

Important: For the Mikrotik hAP ac family of devices, disconnect the Ethernet cable from the WAN port (1) on the Mikrotik and insert it into one of the LAN ports (2,3,4) before you proceed.

---

2. You should be able to ping the node at 192.168.1.1. Don't proceed until you can ping the node. You may need to disconnect and reconnect your computer's network cable to ensure that your IP address has been reset. Also, you may need to clear your web browser's cache in order to remove cached pages remaining from your node's previous firmware version.



3. In a web browser, enter the URL `http://192.168.1.1` to display the page for selecting the **sysupgrade** file. Browse to find the *sysupgrade* file you previously downloaded to your computer, select it, and click the `Upload & Reboot` button.

After successfully installing the *sysupgrade* file the node will automatically reboot to the new AREDN® firmware image.

5.5 TP-LINK first install process

These devices may allow you to use the manufacturer's native web interface to apply new firmware images. If available, this is the most user-friendly way to install AREDN® firmware. Navigate to the system setup menu to select and upload new firmware. Check the TP-LINK documentation for your device if you have questions about using their built-in user interface. If this process works then you will have AREDN® firmware installed on your device and you skip all of the steps described below.

If the process above does not work or if you choose not to use the *PharOS* web interface, then you can install AREDN® firmware on your device using steps similar to those described above for Mikrotik devices. TP-LINK devices are programmed to use TFTP for downloading a boot image from an external source. If you already have a PXE *server* on your Windows computer then you can use that. The example below uses *Tiny PXE*. It may also be possible to use a simple TFTP server instead. For more information, see the **Preparing Your Computer** section above.

Install Preparation

- Download the appropriate TP-LINK *factory* file and rename this file as `recovery.bin`
- Set your computer's Ethernet network adapter to a static IP address of 192.168.0.100.
- Connect an Ethernet cable from your computer to the network switch, and another cable from the LAN port of the PoE adapter to the switch. Finally connect an Ethernet cable from the *POE* port to the node, but leave the device powered off for now.

Linux Procedure

1. Create a directory on your computer called `/tftp` and copy the TP-LINK `recovery.bin` file there.
2. Determine your computer's Ethernet interface name with `ifconfig`. It will be the interface you set to 192.168.0.100 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below. Use a `dhcp-range` of IP addresses that are also on the same subnet as the computer: for example 192.168.0.110,192.168.0.120 as shown below.
3. Open a terminal window to execute the following dnsmasq command with escalated privileges:

```
> sudo dnsmasq -i eth0 -u joe --log-dhcp --bootp-dynamic --dhcp-
↪range=192.168.0.110,192.168.0.120 -d -p0 -K --dhcp-boot=recovery.
↪bin --enable-tftp --tftp-root=/tftp/
```

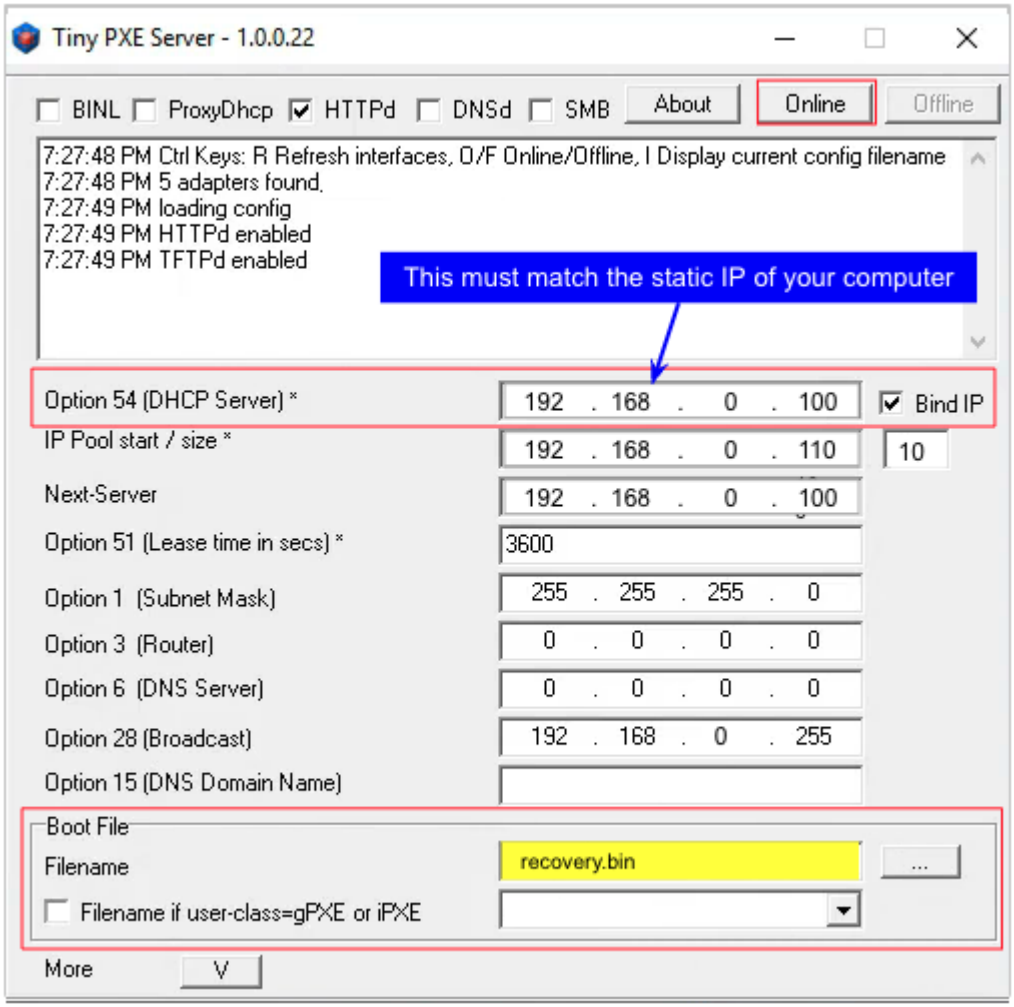
4. With the unit powered off, press and hold the reset button on the radio while powering on the device. Continue to hold the reset button until you see output information from the computer window where you ran the dnsmasq command, which should happen after 20-30 seconds. Release the reset button when you see the "sent" message, which indicates success, and you can now <ctrl>-C or end dnsmasq.

5. The node will now automatically reboot with the new AREDN® firmware image.

Windows Procedure

Configure the PXE or TFTP Server on your Windows computer. The example below uses *Tiny PXE*. For more information, see the **Preparing Your Computer** section above.

1. Navigate to the folder where you extracted the *Tiny PXE* software and edit the `config.ini` file. Directly under the `[dhcp]` tag, add the following line: `rfc951=1` then save and close the file.
2. Copy the `recovery.bin` firmware image into the `files` folder under the *Tiny PXE* server directory location.
3. Start the *Tiny PXE* server exe and select your computer's Ethernet IP address from the dropdown list called `Option 54 [DHCP Server]`, making sure to check the `Bind IP` checkbox. Under the "Boot File" section, enter `recovery.bin` into the the *Filename* field, and uncheck the checkbox for "Filename if user-class = gPXE or iPXE". Click the *Online* button at the top of the *Tiny PXE* window.



4. With the unit powered off, press and hold the reset button on the node while powering

on the device. Continue holding the reset button until you see `TFTPd: DoReadFile: recovery.bin` in the *Tiny PXE* log window.

5. Release the node's reset button and wait for the image to be transferred to the device. You are finished using *Tiny PXE* when the firmware image has been read by the node, so you can click the *Offline* button in *Tiny PXE*.
6. The node will now automatically reboot with the new AREDN® firmware image.

5.6 GL-iNet first install process

These devices allow you to use the manufacturer's pre-installed *OpenWRT* web interface to upload and apply new firmware images. Check the GL-iNet documentation for your device if you have questions about initial configuration. GL-iNet devices provide DHCP services, so you should be able to connect your computer and automatically receive an IP address on the correct subnet. GL-iNet devices usually have a default IP address of 192.168.8.1, so if for some reason you need to give your computer a static IP address you can use that subnet.

After the GL-iNet device is first booted and configured, navigate to the **Upgrade** section and click *Local Upgrade* to select the AREDN® *sysupgrade.bin* file you previously downloaded to your computer.

> **Warning:** Be sure to **uncheck** the **Keep Settings** checkbox, since GL.iNet settings are incompatible with AREDN® firmware. Also, the AR300M16 devices may have a *boot_dev* switch, so be sure to read the GL.iNet boot documentation to select the correct boot mode.

The node will automatically reboot with the new AREDN® firmware image. If for some reason you need to recover your device to factory firmware, you can search the GL-iNet website for a recovery procedure such as the process documented here: GL-iNet debrick procedure

5.7 Cudy first install process

Cudy Travel Routers use a customized version of OpenWRT software, but the "Update Firmware" option in the Cudy web interface only accepts their RSA-signed images. At the present time Cudy provides a generic signed OpenWRT image so that customers can install another OpenWrt version of their choice. This situation adds an extra step to the normal OpenWRT upgrade process:

1. Login to the stock Cudy web interface and use the "Upgrade Firmware" option to install the intermediate firmware image provided by Cudy to OpenWRT (for example, `cudy_tr3000-v1-sysupgrade.bin`)
2. Login to the newly installed intermediate OpenWRT firmware and use "System > Flash Firmware" to install the appropriate AREDN® firmware image

Check the Cudy documentation for your specific device if you have questions. Several models of Cudy Travel Router are supported (for example: Cudy TR1200 v1 and Cudy TR3000 v1). Since these models use different chipsets, be sure to obtain the correct installation files for your specific model. The example below uses the TR3000 image.

- Download the Cudy intermediate firmware image for your device and save it to your computer. At the present time there is a download link toward the bottom of this Cudy FAQ post. Be sure to download the correct intermediate firmware for your specific model. Unzip this file to extract the Cudy intermediate image as a `sysupgrade.bin` file.
- Download the appropriate AREDN® firmware image for your specific device and save the AREDN® `sysupgrade.bin` file to your computer.
- Power on the Cudy router and connect your computer to the LAN port of the Cudy using an Ethernet cable. Your computer should receive an IP address on the 192.168.10.x network from the Cudy DHCP server. Using a web browser, navigate to `http://192.168.10.1`. Follow the initial step to create an admin password for the router, then click `Exit` to display the Cudy web interface.
- Navigate to **General Settings** and click the *Firmware* button in the left navigation bar. In the *Local Update* section, click the `Browse` button and select the Cudy intermediate firmware image you previously downloaded from their download link. For example: `cudy_tr3000-v1-squashfs-sysupgrade.bin`. Click the `Proceed` button to flash the Cudy intermediate firmware.
- After the device reboots, it will be running the intermediate "unlocked" version of OpenWRT using the stock IP network (192.168.1.x). Your computer will need to reconnect to the device in order to receive a new IP address from the OpenWRT DHCP service. You may now navigate to `http://192.168.1.1` and login using the root username without a password.
- In the OpenWRT web interface, navigate to **System > Backup/Flash Firmware**. In the *Flash Firmware Image* section click the `Flash Image` button. Browse to select the AREDN® firmware you previously downloaded for your specific device, then click the `Upload` button.

> **Warning:** In the "Flash image?" box, be sure you **deselect/uncheck** the box for "Keep settings and retain the current configuration." Existing Cudy/OpenWRT settings are not compatible with the AREDN® firmware.

The node will automatically reboot after installing AREDN®, and you may need to refresh your computer's network interface to receive a new IP address before continuing to the *Firstboot node setup* steps in the **Getting Started** section.

If for some reason you need to recover your device to factory firmware, you can search the Cudy website for a recovery procedure such as the process documented here: Cudy recovery procedure. You connect your computer to the *WAN* port of the Cudy, and the Cudy original firmware must be renamed to *recovery.bin*.

5.8 OpenWRT One first install process

These devices allow you to use the manufacturer's pre-installed *OpenWRT* web interface to upload and apply new firmware images. Check the OpenWRT One documentation if you have questions. This device provides DHCP services, so you should be able to connect your computer to its LAN port to automatically receive an IP address on the correct subnet. *OpenWRT One* devices have a default IP address of 192.168.1.1, so if for some reason you need to give your computer a static IP address you can use the 192.168.1.x/24 subnet.

Important: OpenWRT One devices have a boot mode switch. Be sure the boot mode is set to **NAND** (the normal operational setting) before following the example flashing procedure below.

There are several methods for upgrading OpenWRT. The *OpenWRT One* device also provides a USB upgrade method. The example procedure below uses the OpenWRT web interface (LuCi GUI).

- Cable your computer's Ethernet port to the router's LAN port, which is the `1G` Ethernet port next to the Reset button. Power on the *OpenWRT One* and verify that your computer received an IP address from the device's DHCP service. Verify that you can ping the device at 192.168.1.1.
- Open a web browser and navigate to `http://192.168.1.1`. On a fresh device you can login using the default *root* username with an empty password field. If you have already changed the *root* password, then login using your own password.
- Navigate to `System` > `Backup/Flash Firmware`
- Go to the `Flash new firmware image` section and click the `Flash image` button
- Click `Choose File` to select the AREDN® *sysupgrade.bin* file you previously downloaded to your computer, then click the `Upload` button to upload the new image
- **UNCHECK** the `Keep settings` checkbox, then click `Continue` to flash the AREDN® firmware

The node will automatically reboot after installing AREDN®, and you may need to refresh your computer's network interface to receive a new IP address before continuing to the *Firstboot node setup* steps in the **Getting Started** section.

5.9 MorseMicro device install process

Supported models currently include HaLowLink 1, Heltec HT-HD01 & HT-HD7608, and Alfa Tube-AHM. These devices allow you to use the manufacturer's pre-installed *OpenWRT* web interface to upload and apply new firmware images. This device provides DHCP services, so you should be able to connect your computer to its LAN port to automatically receive an IP address on the correct subnet.

- Cable your computer's Ethernet port to the radio's LAN port. Power on the *MorseMicro* device and verify that your computer received an IP address from the device's DHCP service.
- Open a web browser and navigate to `http://192.168.x.1` where `x` is the subnet on which your computer IP address was provided. On a fresh device you can login using the default username and password shown in your device's instructions. If you have already changed the password, then login using your own password.
- Navigate to `System` > `Backup/Flash Firmware` and go to `Flash new firmware image`
- **UNCHECK** the `Keep settings` checkbox
- Click `Choose File` to select the AREDN® *sysupgrade.bin* file you previously downloaded to your computer
- Click `Flash image` to upload the AREDN® firmware

The node will automatically reboot after installing AREDN®, and you may need to refresh your computer's network interface to receive a new IP address before continuing to the *Firstboot node setup* steps in the **Getting Started** section.

5.10 Zyxel device install process

The Zyxel NWA55AXE allows you to use the manufacturer's native web interface to apply new firmware images. Follow the guidelines in the Zyxel User Guide for connecting the device to the supplied PoE to power it on. Use an Ethernet cable to connect the Zyxel LAN port to your Ethernet switch where your computer is also connected (see diagram at the top of this section).

Check the Zyxel User Guide for the steps for accessing the web configurator. By default the Zyxel device has an IP address of 192.168.1.2, so give your computer an IP address in the 192.168.1.x subnet (for example, 192.168.1.10). Open your web browser and navigate to `http://192.168.1.2` where you will be prompted to enter a username and password. By default the Zyxel `admin` account uses the password that is printed on the label attached to the device. Select your preferred language and choose the **Standalone** management mode.

In the navigation bar on the left, click the *Configuration* icon and navigate to the *Maintenance* menu. Under *File Manager* select *Firmware Package*. To upload the AREDN® firmware, browse to the appropriate *FACTORY* image you previously downloaded from the AREDN® Firmware Selector,

then click the *Upload* button. The node will automatically reboot with an IP address of 192.168.1.1 so that you can configure the AREDN® firmware.

The OpenWRT information page for this device may provide additional help for more in-depth troubleshooting.

5.11 After the AREDN® firmware install

After the node reboots, it should have a default IP address of 192.168.1.1. Make sure your computer has an IP address on the 192.168.1.x network. You should be able to ping the node at 192.168.1.1. Don't proceed until you can ping the node. You may need to disconnect and reconnect your computer's network cable to ensure that it has a connection.

Once your device is running AREDN® firmware, you can display its web interface by navigating to either `http://192.168.1.1` or `http://localnode.local.mesh`. Some computers may have DNS search paths configured that require you to use the fully qualified domain name (FQDN) to resolve *localnode* to the mesh node's IP address. You may need to clear your web browser's cache in order to remove any cached pages.

You can use your web browser to configure the new node with your callsign, admin password, and other settings as described in the **Firstboot Node Setup** section of the documentation.

5.12 Node Reset button actions

The reset button on an AREDN® node has two built-in functions based on the length of time the button is pressed. This may be helpful if you need to recover a lost *admin* password, or if you want to reconfigure you node by starting with a fresh "just flashed" state.

With the node powered on and fully booted:

- To reset only the node admin password and DHCP service, hold the reset button for **5 seconds**. The default *admin* password is `hsmm`.
- To reset a node to "firstboot" state, hold the reset button for **15 seconds**.

On some equipment models it may be possible to accomplish these reset procedures by pressing the *Reset* button on the PoE unit.

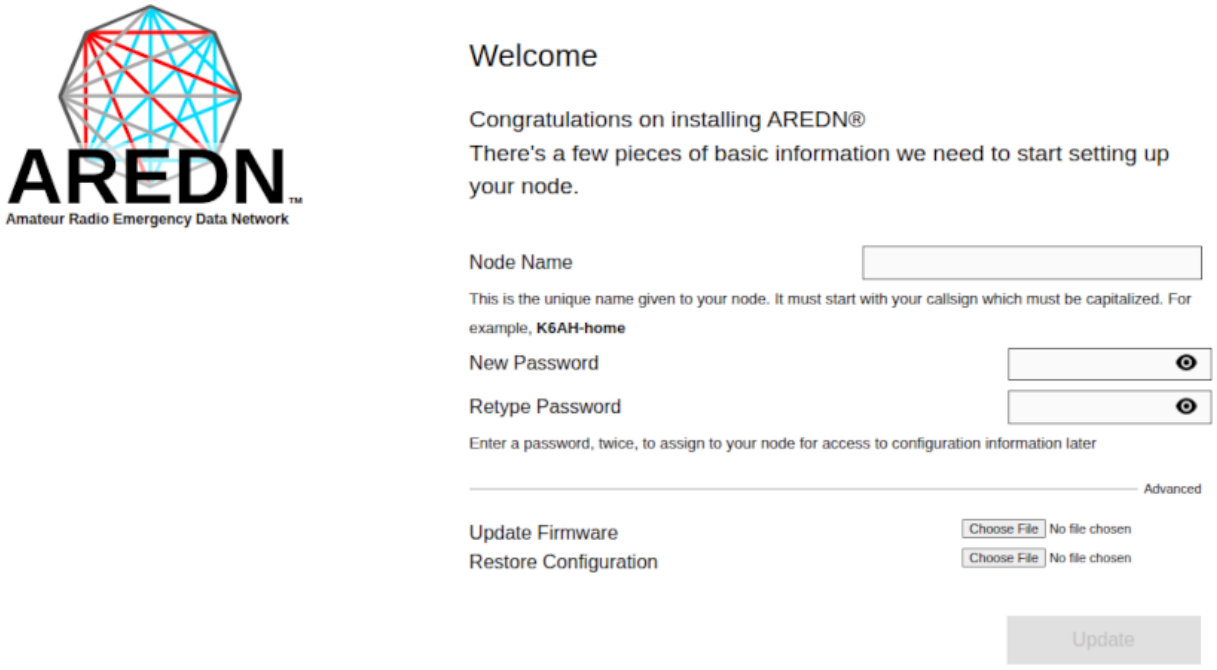
CHAPTER

SIX

FIRSTBOOT NODE SETUP

After you have installed the AREDN® firmware and rebooted the device, the node will have a default IP address of 192.168.1.1. You can set your computer to receive an IP address from your node via DHCP. After connecting your computer to a LAN port on the node or the PoE unit, you should be able to ping the node at 192.168.1.1. Navigate to your node's web interface at `http://192.168.1.1` or `http://localnode.local.mesh`. Some computers may have DNS search paths configured that require you to use the fully qualified domain name (FQDN) to resolve *localnode* to the mesh node's IP address.

The firstboot status page will be displayed, instructing you to configure your node by entering a node name and password for administrative access to your node.



Node Name

Begin the node name with your CALLSIGN in all capital letters followed by a dash character and some unique identifying information of your choice. Node names may contain up to 63

letters, numbers, and dashes, but cannot begin or end with a dash. Underscores, spaces, or any other characters are not allowed. Amateur radio operators are required to identify all transmitting stations, so your node name is beaconed automatically by the node every five minutes. Recommended names follow the (CALLSIGN)-(label) format, such as AD5BC-MOBILE or AD5BC-120SE. As a general rule node names should be kept as short as possible, while clearly and uniquely identifying the node.

Password
Set a new administration password for the node. Typically passwords may contain the characters `a-z`, `A-Z`, `0-9`, period `.`, dash `-`, underscore `_`, exclamation `!`, and tilde `~`.

Note: Avoid Linux reserved characters, including but not limited to: `# $ & * < > % \`

Enter your new password again in the *Retype Password* box to verify it is correct. You can click the *eye* icon at the right of the password fields to toggle between hidden and visible text. The first time a node is configured it will require you to set the password. Be sure to remember or record the password so you can use it for any future administrative tasks on the node.

After providing the new node name and password, click the *Save & Reboot* button. Once your node reboots it will have an IP address in the 10.x.x.x range, so you should set your computer to use DHCP to obtain a new IP address from your node. You may need to disconnect/reconnect or disable/enable your computer's Ethernet interface so that it connects using the new IP address.

6.1 Advanced Options

The firstboot display also has an **Advanced** section which you can view by clicking the *Advanced* label. These are helpful when you have an existing node that has been reset to *firstboot* state, as described under the next heading in this document.

- If you want to update your node to a specific version of firmware, you can choose that firmware file and click the **Update** button.
- If you previously saved your node's settings in a *backup* file, you can apply those saved settings by choosing the *backup* file and clicking the **Restore** button.

6.2 Resetting a node to *firstboot* state

There are two ways to reset an existing node to its *firstboot* state, which allows you to start fresh to reconfigure a node.

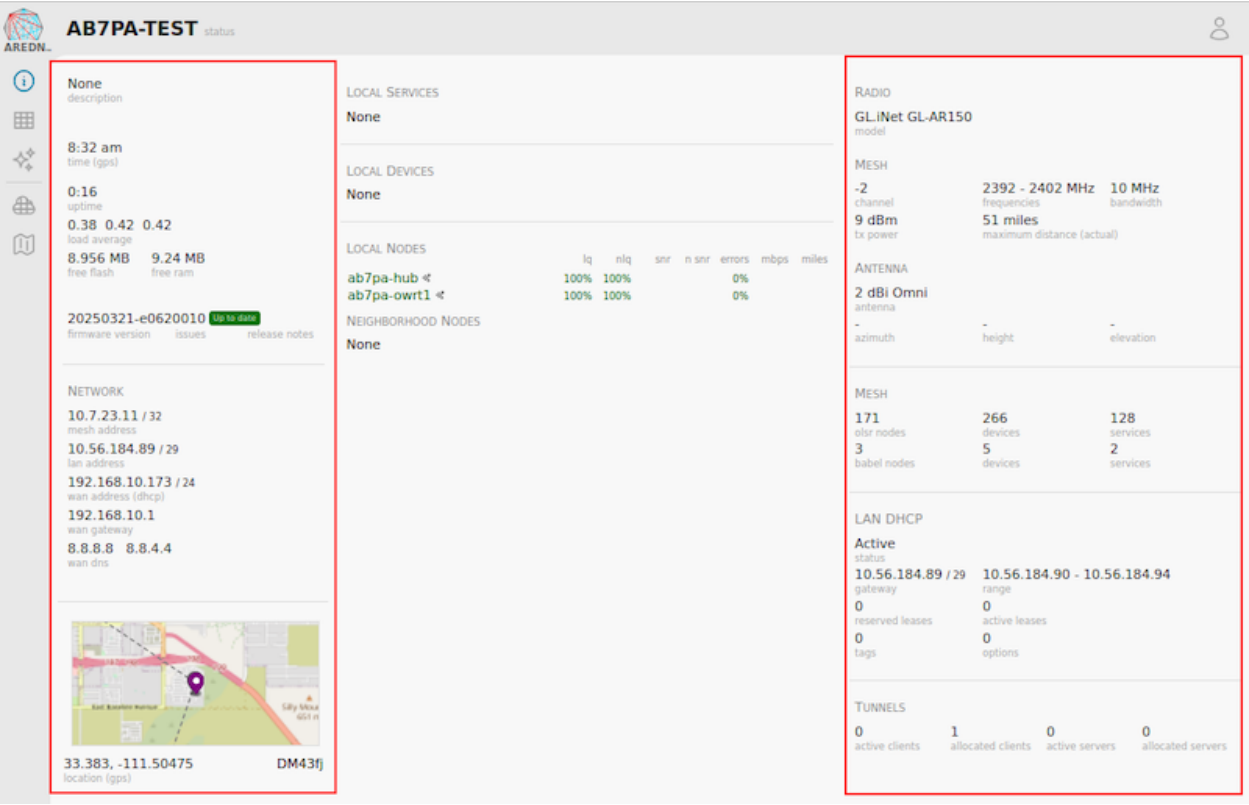
1. In *admin* mode you can navigate to the firmware section to turn off the *Keep Configuration* switch under **Advanced Options**, then click `Done` and `Commit` your change. Now you can reinstall the existing firmware version on your node, at which time you can configure the node from its *firstboot* state.
2. With your node powered on and running its current AREDN® firmware, press and hold the reset button on your node for 15 seconds. This will cause your node to enter its *firstboot* state from which you can start fresh with your node's configuration.

CHAPTER

SEVEN

NODE STATUS DISPLAY

Once you have completed the initial setup on your AREDN® node, you can connect your computer to a LAN (Local Area Network) port on the device or the PoE and use a web browser to navigate to the **node status** page. `http://localnode.local.mesh` or `http://<your-nodename>.local.mesh`

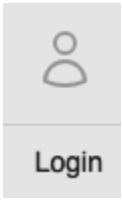


This display has been designed to present all of the important information about your node in one view. Someone navigating to your node's status display will be able to see all of the key elements of

interest without having to click to multiple pages. This display consists of a top navigation bar, a left navigation bar, and typically shows three columns of information about your node. The information is displayed adaptively based on the width of the browser or screen you are using. If you are using a narrow display such as a smartphone or tablet, the information may be shown in a single column. If you are using a width-limited display, the information may be shown in two columns. On wider displays the information will be shown in three columns (as depicted above).

7.1 Top Nav Bar

The AREDN® logo is displayed in the upper left corner. If you click the AREDN® logo you will be taken back to `localnode.local.mesh`. After the AREDN® logo, the node name is displayed along with a label indicating whether you are viewing the *status* or *admin* display. On the far right is the user status icon.



The default icon indicates that you are viewing the page as a normal user. Clicking this icon will open a dropdown menu that allows you to login as the node administrator.

If the user icon is displayed within a circle, then you are viewing the page as the node administrator. Clicking this icon will open a dropdown menu that allows you to logout, reboot the node, display help, or open the AREDN® website.

7.2 Left Nav Bar

Using the icons on the left side bar you can navigate to various displays.

 navigates to this **node status** display.

 navigates to the local mesh status page showing the nodes visible on the local mesh network, as well as any services provided by those nodes.

 navigates to the *Cloud Mesh* view through the Supernode network (if your local network has a Supernode). For more information, see **Supernode Architecture** in the *Network Topologies* section of the **Network Design Guide**.

 navigates to the world map on the AREDN® website. This is only displayed if your node has valid latitude and longitude values, since this feature is designed to display your node in the center of the map. If your node has no latitude and longitude values, then this icon will not appear on the nav bar.

7.3 Left Section

Several sections of node information are presented here (listed from top to bottom).

Node Description
: This is not a required field, but node owners typically use it to list their contact information or the tactical purpose for the node.

Node Time, Uptime, Load Average, and Free Memory
: The node time is displayed and if an NTP or GPS time source is available, that source will be displayed in parentheses. If an Internet connection or a local NTP server is available, your node's NTP client will sync its time with that source. If a local GPS source is available, your node can use that source.

  The `uptime` is also displayed, which is the time since the last reboot. `load` is the average system utilization for the last 1, 5, and 15 minutes. `free flash` and `free ram` shows how much storage space is remaining on your node. `flash` is the internal non-volatile storage where the operating system, configuration files, and software packages are kept. `ram` is the amount of RAM (Random Access Memory) available for running processes on the node.

Firmware Information
: This displays the node's current firmware version. A badge on the right indicates the status of the firmware, with valid values including `Up to date`, `Update available`, and `Custom`.

If your node has access to the Internet you can also click the *issues* label below the firmware version, and this will open the AREDN® Issues page on GitHub. Clicking the *release notes* label will open the Changelog page on the AREDN® website.

Network Information
The Mesh IP address/netmask is displayed using CIDR notation, followed by the LAN IP address/netmask. If the WAN (Wide Area Network) interface is enabled, the WAN IP address/netmask is displayed along with whether this address was obtained via DHCP or assigned as a static IP address. The WAN gateway IP address is also displayed.

Node Location Information
At the bottom of the left column is the node location information. Initially there will be no location values since the latitude, longitude, and grid square have not yet been entered. After the latitude, longitude, and grid square have been entered (as described in the **Node Admin** guide), your node will attempt to display a thumbnail map with its location in the center. If your node has no access to the Internet or to a local map tile server, then the map will not be displayed. The latitude, longitude, and grid square values will be shown below the map thumbnail.

7.4 Center Section

The center column has four main sections (listed from top to bottom).

Local Services
This section displays the service links for any mesh services on your node or its locally-connected devices. These service links are displayed side by side in two columns. Clicking any of the links will navigate to the selected service.

Local Devices
This section displays any devices that are directly connected to your node. This includes devices that are connected to your node's LAN via Ethernet cable (such as VoIP (Voice over IP) phones, IP cameras, or service computers). Be aware that DHCP devices with *Do Not Propagate* checked will not be displayed.

Local Nodes
This section displays any local DtD (Device to Device) nodes that are directly connected to your node, typically via Ethernet cable. If you hover the cursor over the node name, a popup will appear showing the relative link quality of the connection to that node. Clicking the node name will navigate to that node's status page. For *Local Nodes* the snr, nsnr, mbps, and distance columns will always be blank.

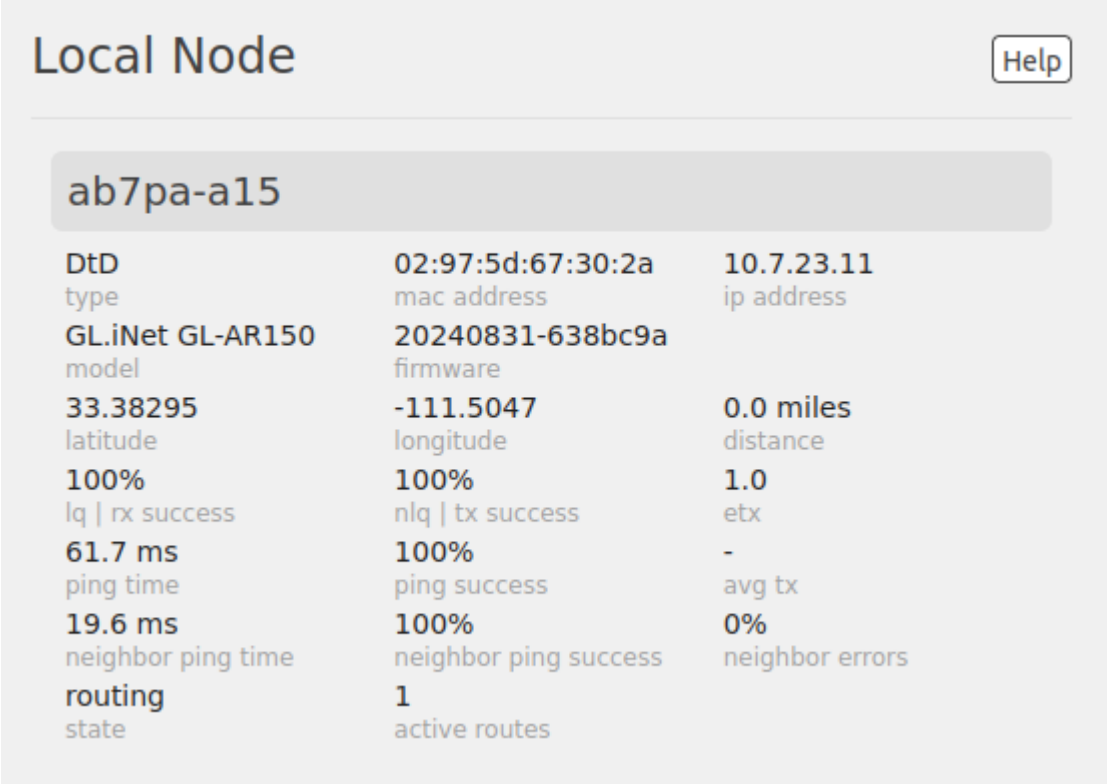
Basic Link Quality Metrics

Several link quality statistics can be displayed for different types of connections that are mentioned below. Before introducing those link types, here is a brief explanation of the link

quality metrics that may be displayed if they are available.

- `rx` or receive success rate shows the percent of packets received based on what was expected.
- `rtt` is the round trip time or "two-way delay" which shows the link latency between two nodes.
- `snr` or Signal-to-Noise Ratio is expressed in decibels (dB). This metric only applies to RF links and represents the level of signal which is detected above the noise floor. *SNR* is shown for both sides of a radio link (SNR (signal to noise ratio) and N SNR (neighbor signal to noise ratio)).
- `errors` is calculated as the moving average of (total sent packets) divided by (total sent packets plus retransmissions) and expressed as a percent. For example, if the node had to send every packet twice for it to be successfully received, the error rate would be 50%. An additional penalty is subtracted if the neighbor node is unpingable.
- `mbps` is a rolling average of the data rate achieved across any radio (RF) link. This column may show zero if the data being transmitted between these nodes is not sufficient for the metric to be calculated.
- `dist` is the line of sight distance between your node and the remote node, calculated from the GPS coordinates if they are entered for both nodes. This value will be expressed in *miles* or *kilometers* based on the locale settings in the web browser.

When you hover over the row of any Local Node, a gray background appears which indicates that row is selected. If you click in the selected row (but not directly on the node name link), the **Local Node** popup will be displayed which provides more detailed information about your node's connection to the selected node.

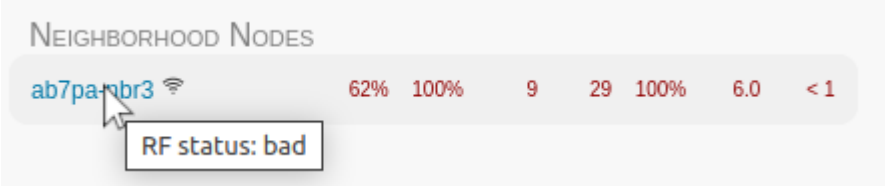


Neighborhood Nodes

This section displays any nodes that are direct neighbors of your node, whether via RF (radio frequency), an xlink, or a tunnel over an Internet connection. Each type of connection will display a different icon to the right of the node name, and this indicates the type of link (for example, the small radio signal icon in the screenshot above indicates an RF link). If a node is reachable from your node, you can click the node name to navigate to that node. Not all the columns for link quality statistics will be populated for *Neighborhood Nodes*. If you hover the cursor over the node name, a popup will appear showing the relative link quality of the connection to that node.

Node Status Indicators

For nodes in the *Local Nodes* and *Neighborhood Nodes* sections, different colors, styles, and hover text may be displayed based on the quality of the connection to each node.



In the example above, the node name and statistics are displayed in red, and hovering over the node name displays status text indicating that the RF status is "bad". Status colors may vary based on the display theme you have chosen. The following list shows how the standard theme text colors are mapped to link quality.

green
: excellent

dark green
: good

blue
: okay

orange
: poor

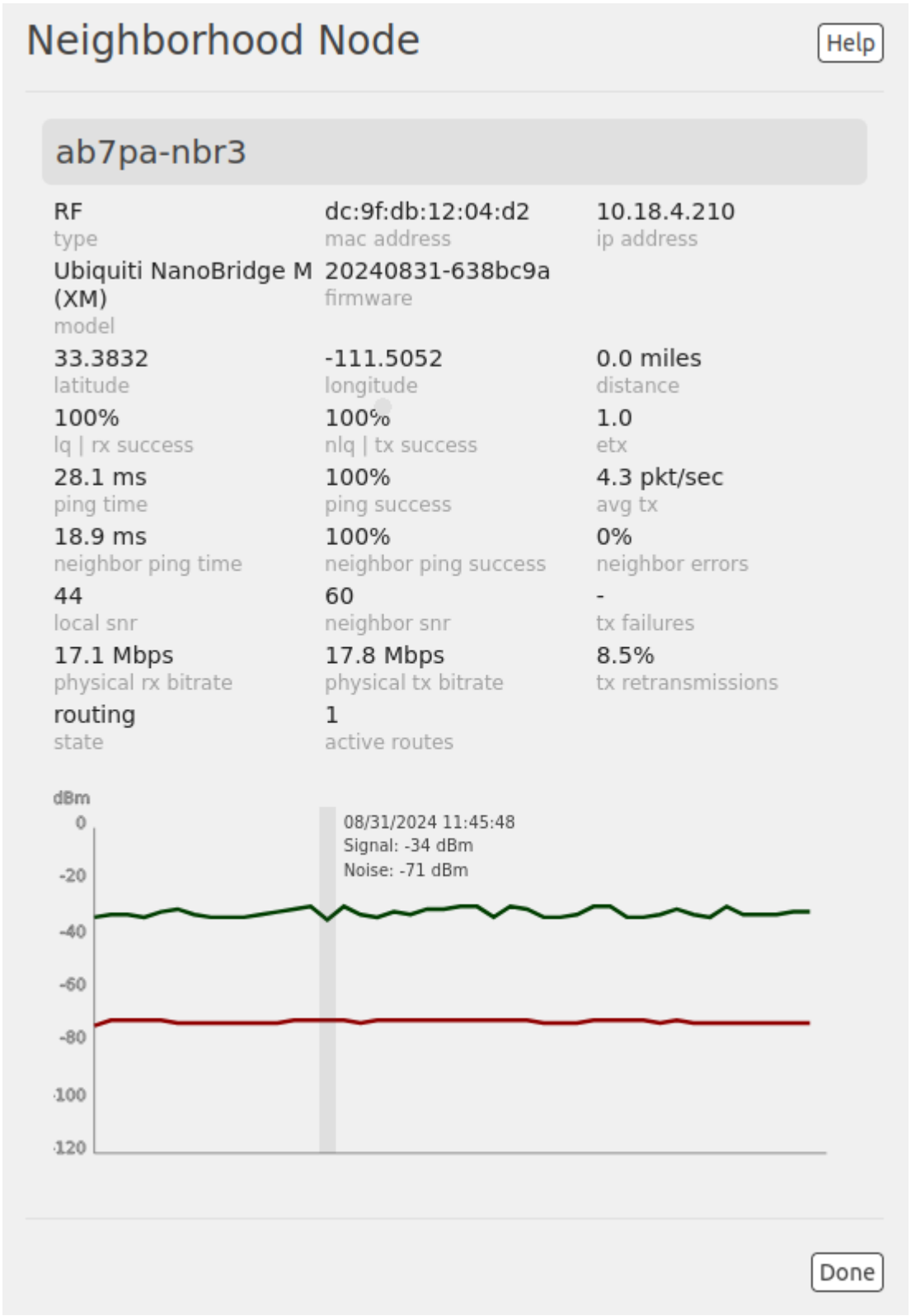
red
: bad

gray
: idle or hidden

strikethrough
: blocked

If hidden nodes are detected which your node cannot reach for some reason, they will be displayed in a subsection labeled *Hidden Nodes*. For more information about hidden nodes, see *Hidden and Exposed Nodes* in the **Channel Planning** section of the **Network Design Guide**.

When you hover over the row of any Neighborhood Node, a gray background appears which indicates that row is selected. If you click in the selected row (but not directly on the node name link), the **Neighborhood Node** popup will be displayed which provides more detailed information about your node's connection to the selected node.



Node Icons

Each node in *Neighborhood Nodes* list may have an icon that indicates its status.

 indicates an RF link

 indicates a tunnel link

 indicates a DTD link

 indicates a Cross-link

7.5 Right Section

The right column displays additional details about your node (listed from top to bottom).

Radio section

Your device manufacturer and model are displayed at the top of the column. If a radio is configured as a `Mesh` radio, you will see the channel number and frequency range, followed by the channel width (in MHz (Megahertz)). Next is the transmit power (in dBm (decibels in millivolts)), the actual maximum distance limit (in miles or kilometers) calculated by the radio, and the minimum SNR (in dB (decibels)) for communication with other RF nodes.

Note: The `maximum distance (actual)` value is pulled from the radio's kernel, so it is the real value in use by the radio. Some radios, typically the indoor desktop devices, do not allow this value to be changed and will always report a fixed value or possibly zero.

If a radio is configured as a `LAN Hotspot`, you will see the channel number and the SSID that wifi clients can use to connect to your node's hotspot. If a radio is configured as a `WAN Client` you will see the SSID of the wifi AP to which your node connected, as well as the status of the connection (no connection, connected but no internet, connected with internet). Your node's antenna information is listed next, including the type of antenna, the azimuth, height above ground level, and tilt angle / elevation (if directional).

Mesh section

Next there are summary statistics showing how many nodes, devices, and services are currently visible from this node.

LAN DHCP section

By default each node runs a DHCP server which is capable of automatically providing IP addresses for any LAN-connected devices. This section shows the DHCP (Dynamic Host Configuration Protocol) server status, the IP address/netmask of your node functioning as the gateway for its LAN-connected devices, and the IP address range served by your node. It also shows the number of active leases and IP addresses reserved for specific devices on its LAN. In addition, counts are displayed for the number of DHCP tags and options that are defined on your node.

Ethernet Ports & Xlinks

If you have a multiport node or one which supports xlinks, then the *Ethernet Ports & Xlinks* section will be displayed. This shows the number of Ethernet ports on the device, as well as how many of them are actively in use. You will also see the number of xlinks that you have defined on this node.

Tunnels section

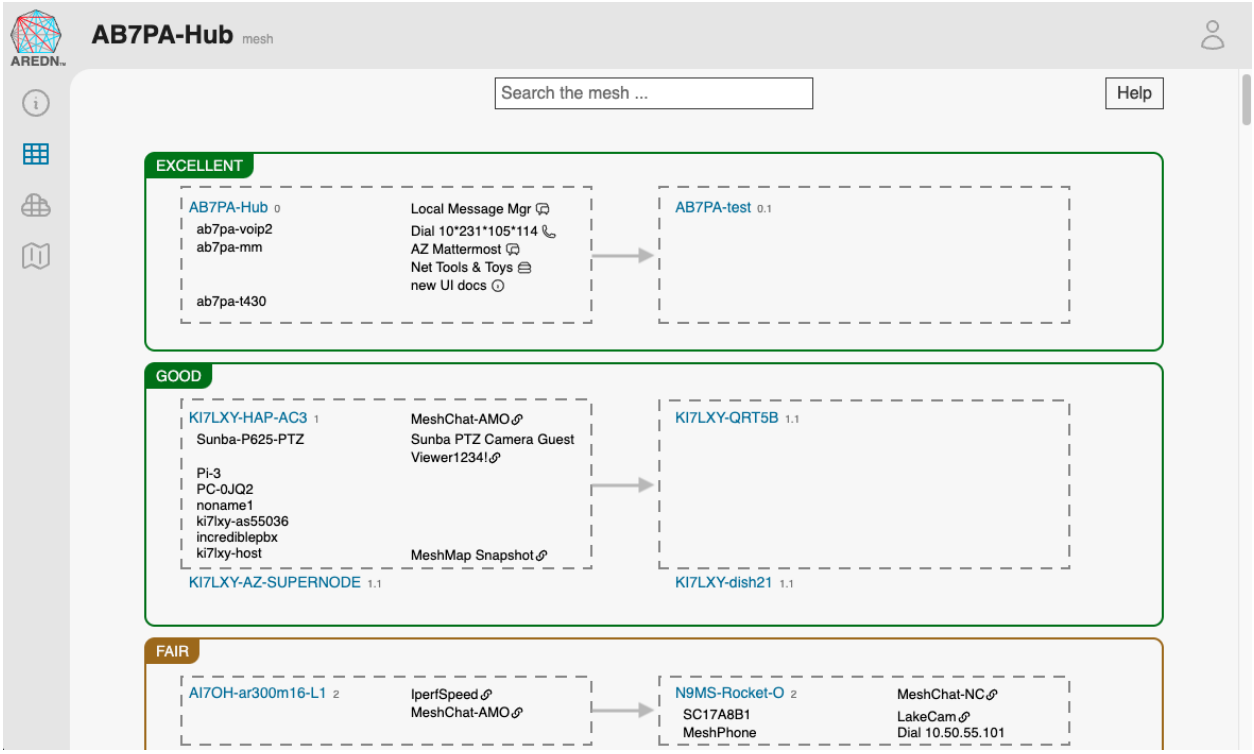
This section displays statistics on any tunnel connections you may have on your node. Counts are displayed for active / allocated tunnel client connections as well as for active / allocated tunnel server connections on your node.

CHAPTER

EIGHT

MESH STATUS DISPLAY

You navigate to the **mesh status** page by clicking the mesh icon in the left nav bar.



At the top of this page there is a search box which allows you to filter the mesh network display to include only those nodes, devices, and services which match the keywords you enter. As you type each character from your keyboard into the search field, the display will change to show only the entries that match your character string. The filter is case insensitive, so it will find both uppercase and lowercase entries for the characters you enter. To restore the original display, delete your search

characters or refresh the page in the web browser. To the right of the search field there is a `Help` button which explains the use of the **mesh status** page.

The **mesh status** page is divided into several groups of devices based on the link quality. The top groups are more likely to be reachable by your node than are the devices in groups toward the bottom of the page.

Within each group the nodes are displayed side by side in two columns. The node in the upper left will have the best link quality, followed by the next best node to its right, then continuing down each row of the display. Hovering the cursor over the left or right column will display a gray background, making it easy to see which node you are focused on. Clicking the node name will navigate to the node status display of that node.

Each node will show the node name followed by a number that represents the route cost to that node, with lower numbers representing better links. Nodes are put into groups based on their route cost. The display shows each node, any connected LAN devices, and any advertised services available on the node and its hosts. Small icons appear at the right of each service which indicate the type of service, and the definition of these service types is described in the *Local Services* section of the **Node Admin** guide. You can click any available web links to navigate to the nodes or services shown on the **mesh status** display.

CHAPTER

NINE

NODE ADMIN GUIDE

You must login as the node administrator in order to perform node management tasks.

 Click the user icon at the far right of the top nav bar. Select `login` and enter your node's admin password (which was configured when you installed the AREDN® firmware).

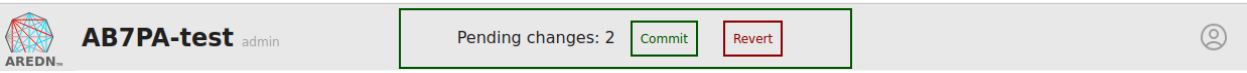
 Upon successful authentication you will see the admin icon, and the label to the right of your node name should say *admin*.

9.1 Admin navigation & actions

In *admin* mode the sections on the **node status** display become editable and new sections with additional options will appear. When you hover the cursor over a section, a gray background appears which identifies that section as being configurable. When you click in a highlighted section, a new settings dialog display will be opened. The title at the top of the display tells you what settings you are configuring. There is also a `Help` button in the upper right corner which will enable extended context-sensitive descriptions of each option which has additional help text.

Settings can be edited or selected from dropdown lists by clicking in each of the fields. If a section has **Advanced Options** you can view and configure them by clicking the *Advanced Options* label to display those additional settings. After making any changes to the configuration settings on each display, you will typically click the `Done` button. Your changes have been recorded but they have not yet been committed or saved to your node. You may also click the `Cancel` button to discard any changes you have made and return to the *admin* view.

After clicking `Done` you will be returned to your node's *admin* view where you will see a new item in the top nav bar. Click the `Commit` button to apply your change(s) or the `Revert` button to ignore any change(s) and revert to the previous settings.

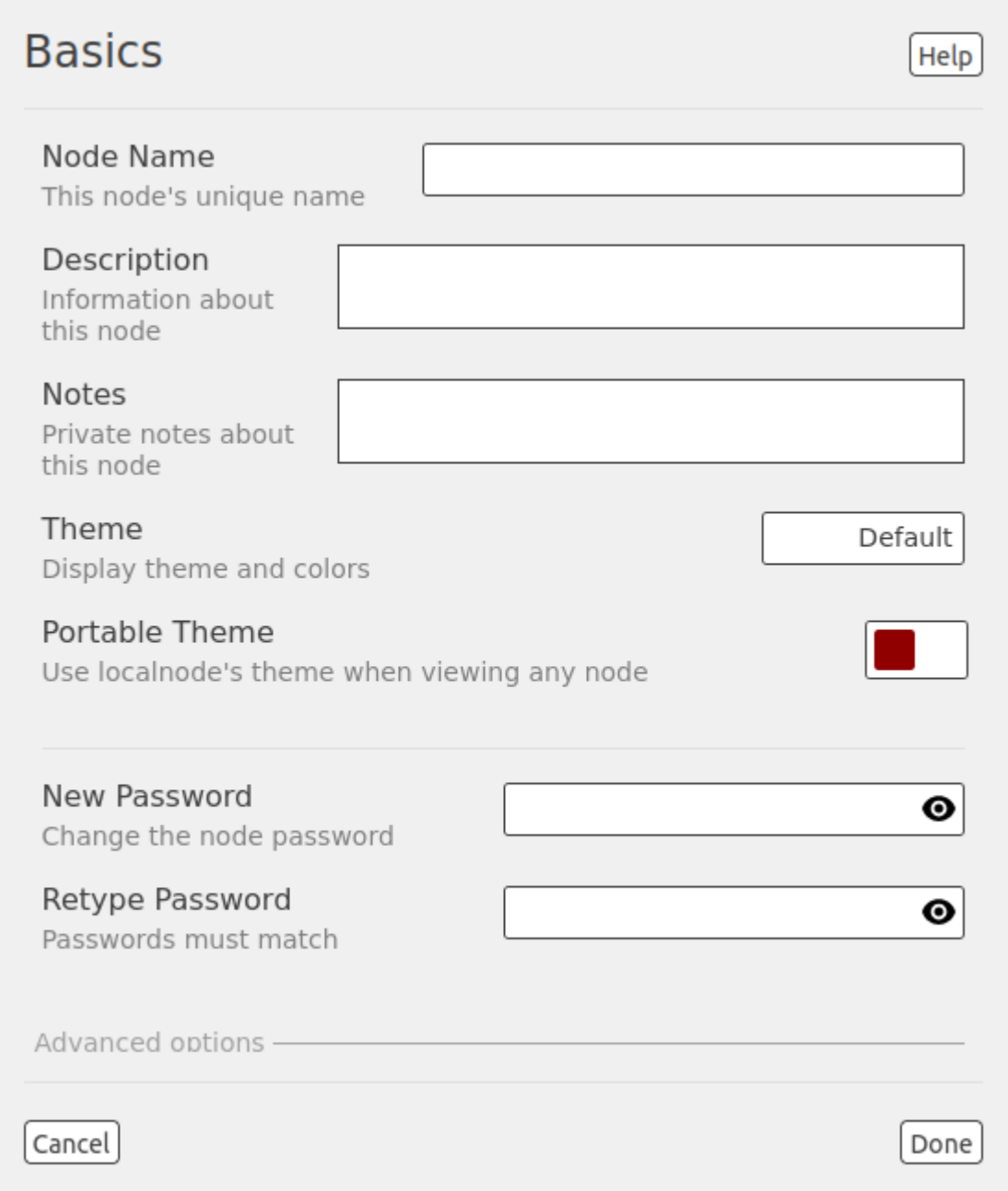


For some configuration changes there may be additional action buttons that are displayed. For example, if you want to upload or remove an SSH security key you will press the `Upload or Remove Key` button. Or you can press the `Fetch and Update` button to install a firmware image, or press the `Remove` button to remove a package installed on your node. In some cases you may need to scroll down on the configuration display in order to see these buttons.

The sections of the **admin** display will be described below, beginning in the upper left corner of the left column and working down that column before moving to the center and right columns of the display, working from top to bottom on each column.

9.2 Basics

Starting at the top of the left column, highlight and click the section which contains the description and notes. This *Basics* section allows you to configure the following settings. Context-sensitive help is available by clicking the `Help` button.



Node Name

Begin the node name with your CALLSIGN in all capital letters followed by a dash character and some unique identifying information of your choice. Node names may contain up to 63 letters, numbers, and dashes, but cannot begin or end with a dash. Underscores, spaces, or any other special characters are not allowed. Amateur radio operators are required to identify all transmitting stations, so your node name is beaconed automatically by the node every five minutes. Recommended names follow the (CALLSIGN)-(label) format, such as AD5BC-MOBILE or AD5BC-BLACKMTN. As a general rule node names should be kept as short as possible, while clearly and uniquely identifying the node.

Description

This is not a required field, but it is a good place to describe the features or function of this device. Many operators use this field to list their contact information or the tactical purpose for the node. If you want to display information about your node, put that information here in the description rather than making it part of the node name. There are no character restrictions in this field, but the maximum length is 210 characters.

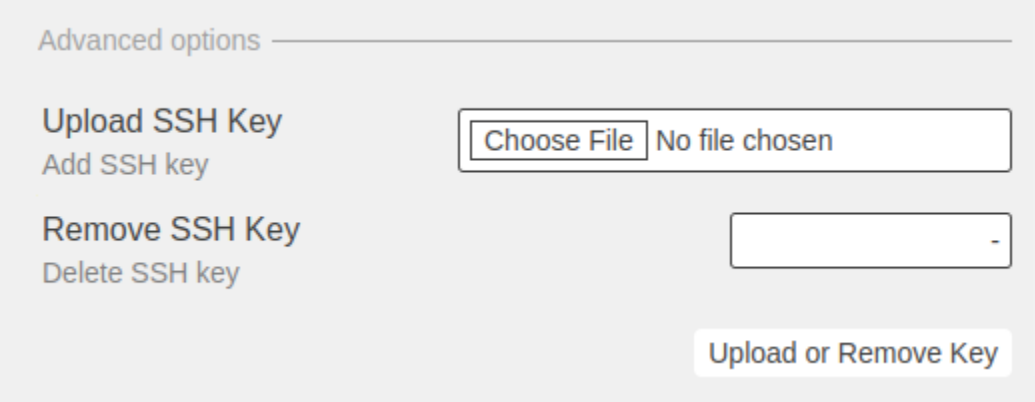
Notes
This optional field allows you to enter notes about this node which are only visible to the node admin. For example, you may enter information about special settings or configurations for links to nearby devices.

Theme
Click in the field at the right to select a theme from the dropdown list. Your node will immediately display your page in the selected theme.

Portable Theme
This switch allows you to enable theme portability. When *Portable Theme* is enabled, the theme on your localnode will be the theme used when viewing any node on the mesh. By default this setting is disabled, which means that the remote node owner's theme will appear in your browser.

Password
Typically passwords may contain the characters `a-z`, `A-Z`, `0-9`, period `.`, dash `-`, underscore `_`, exclamation `!`, and tilde `~`. Avoid Linux-reserved characters, including but not limited to `#`, `$`, `&`, `*`, `<`, `>`. Enter the new password again in the *Retype Password* box to verify it is correct. You can click the *eye* icon at the right of the password fields to toggle between hidden and visible text. Be sure to remember or record the new password so you can use it for any future administrative tasks on the node.

Additional options will be displayed when you click **Advanced Options**.

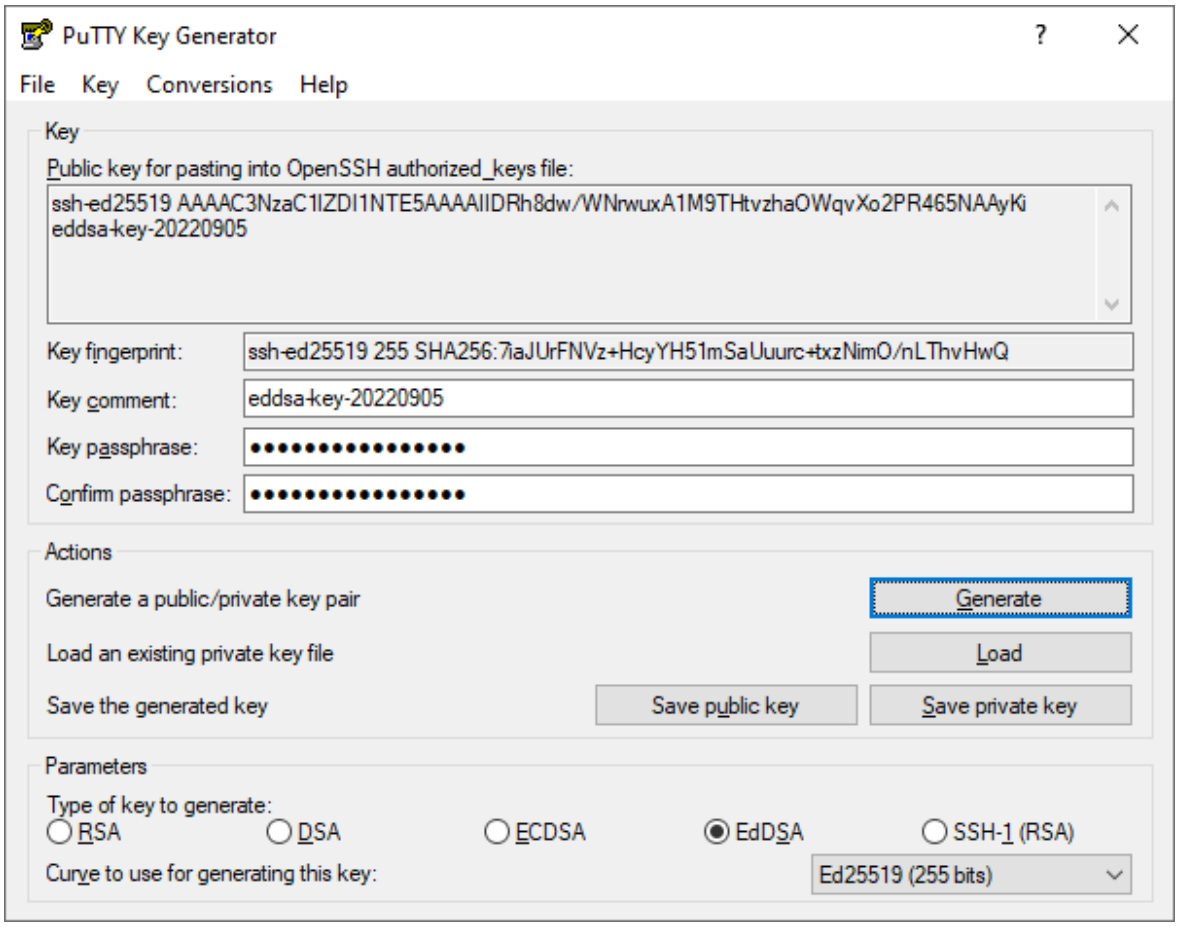


9.2.1 SSH keys

Upload SSH Key

Uploading SSH keys allows computers to connect from the command line to your node via SSH without having to know the password. SSH keys are generated on your local computer using native utilities such as *ssh-keygen* or the PuTTY program's *Key Generator*. Currently AREDN® nodes support RSA, ECDSA, and Ed25519 key types. RSA is the most widely supported type and is generally considered secure with a minimum key size of 2048 bits. Ed25519 is more recent and has a smaller key size (255 bits), but older devices using the *tiny-build* firmware do not support Ed25519 keys.

If you plan to generate ssh keys on a Microsoft Windows computer, you may want to review the PuTTY documentation which describes this process. The example below shows the generation of an Ed25519 key using the PuTTY Key Generator.



Once you generate the key files on your computer, you can upload the *public* key to your AREDN® node. Click the `Browse` button to locate the *public* key file, then click the `Upload Key` button at the lower right.

Remove SSH Key

To remove an existing SSH key, click in the field at the right and select the key from the

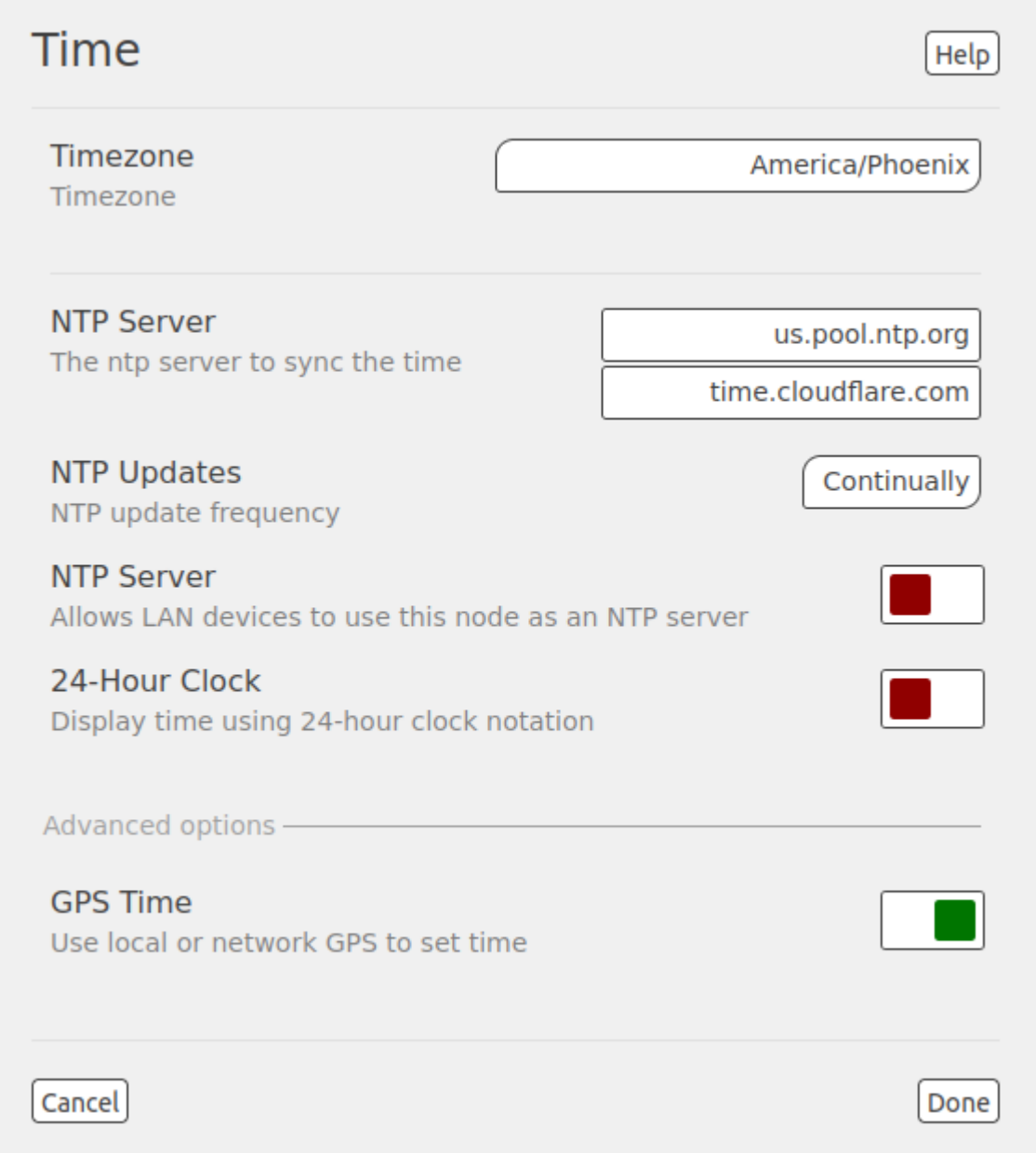
dropdown list. Then click the `Remove Key` button at the lower right.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

9.3 Time settings

Highlight and click the section displaying your node's time. Select your timezone from the dropdown list, where the default value is UTC (Coordinated Universal Time). Two fields are provided for entering the hostnames of NTP (Network Time Protocol) servers if your node is connected to a network with network time services. You can enter valid hostnames in the *NTP Server* fields: for example `us.pool.ntp.org` or `AD5BC-ntp.local.mesh`. You may also choose how often NTP will update the node's clock by selecting a value from the *NTP Updates* dropdown list. The default is once per day [Daily] but you may also select once per hour [Hourly] or you can have your node run the NTP program [Continually].

If you run NTP on your node *Continually* then a new switch will appear which allows your node to function as an NTP Server for any of your LAN-connected devices. The *NTP Server* switch is `disabled` by default.



By default the time on your node will be shown as a 12-hour clock with *am/pm*. To display node time using a 24-hour clock, enable the *24-Hour Clock* switch.

Additional options are displayed when you click **Advanced Options**.

By default your node can use a local **GPS Time** source if one is available. To disable this behavior, slide the switch to the *off* position.

If you want your node to function as a GPS time *server* you will need to install the `whenandwhere`

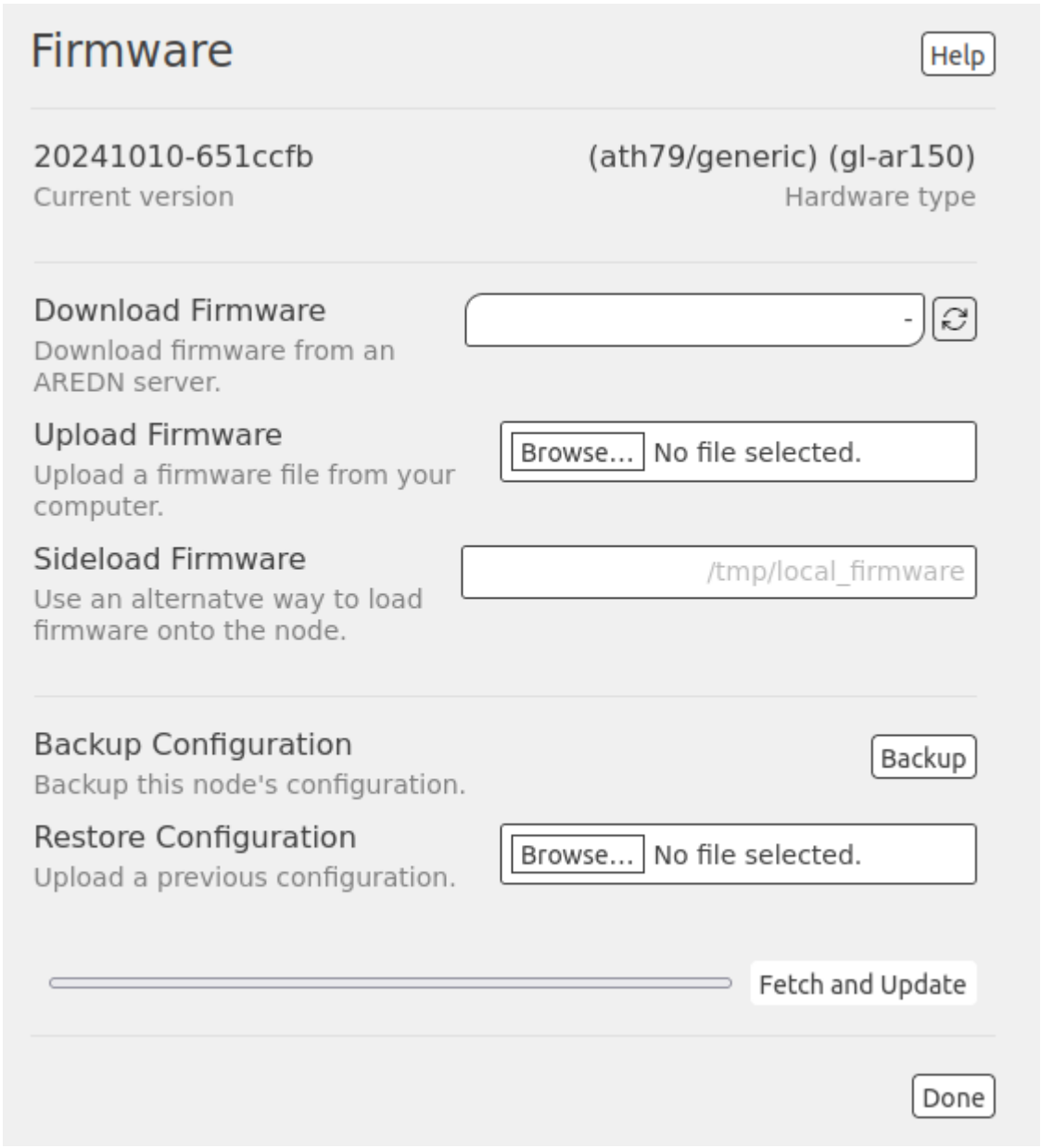
package. Select and install this package from the *Packages* > *Download Package* list on your node. This meta-package installs the required dependencies for a USB GPS dongle so your node can provide GPS time for itself and other local DtD linked devices.

If you plan to use Wireguard tunneling, make sure that a GPS or NTP time source is reachable when the node boots so that the key exchange between the client and server will happen correctly. Without proper time synchronization, Wireguard will not establish tunnels.

Context-sensitive help is available by clicking the `Help` button. You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

9.4 Firmware settings

Highlight and click the section displaying your node's firmware version. The top field displays the currently installed version of firmware on your node. Context-sensitive help is available by clicking the `Help` button. There are three ways to update your node's firmware.



9.4.1 Download Firmware

If your node has Internet access or access to a firmware server on your local network, you can click the *refresh* icon on the right side of the field in order to update the list of available images. Select the image to install and click the `Fetch and Update` button to begin the process. You may need to scroll down in the display to see the `Fetch and Update` button.

9.4.2 Upload Firmware

If you have a new firmware image that you already downloaded to your local computer from the AREDN® website or a local firmware repository, click the `Browse` button and navigate to the location where you saved the firmware file. Select the image to install and click the `Fetch and Update` button to begin the process. You may need to scroll down in the display to see the `Fetch and Update` button.

9.4.3 Sideload Local Firmware

If you need to remotely upgrade the firmware on a node which has a marginal connection to the network, the standard web/http method may not reliably transfer the image to the node. In this situation you may want to use an independent means of uploading the firmware to the node before beginning the upgrade process. Choose an upload method such as `scp` (secure copy) with a long connection timeout, which may allow the file transfer to continue the upload in the event of a network interruption. Transfer the new firmware file to your node, place it in the `/tmp` folder, and name it `local_firmware`. Once the node detects the presence of `/tmp/local_firmware`, then the filename in the field at the right will become active. Click the `Update` button to begin the process. You may need to scroll down in the display to see the button.

9.4.4 Backup Configuration

Once you have your node configured the way you want it, you can save those configuration settings by clicking the `Backup` button on the *Firmware* menu. This will create a compressed archive of the node's configuration files and it will download the timestamped backup file to your local computer. This snapshot file can be used to restore your node's configuration to a known good point in time, and it also allows you to transfer the configuration to another device using the same hardware.

It is also possible to generate a node backup file from the command line of your node:

```
# /usr/local/bin/backup
Generated backup file: /tmp/node-backup.backup
```

> **Attention:** A saved backup file can only be used to restore to the same node or identical hardware running the same firmware version, but it cannot be restored to different hardware or a node running a different firmware version.

The backup file will include all of the information contained in the files and directories listed in the `/etc/arednsysupgrade.conf` file. For example, this currently includes:

```
/etc/config.mesh/setup
/etc/config.mesh/wireguard
/etc/config.mesh/aredn
/etc/config.mesh/xlink
/etc/config.mesh/firewall.user
/etc/config.mesh/dropbear
/etc/aredn_include/swconfig.user
/etc/aredn_include/static_routes
/etc/aredn_include/fixedmac.lan
/etc/aredn_include/fixedmac.wan
/etc/aredn_include/fixedmac.dtdlink
/etc/aredn_include/bridge.network.user
/etc/aredn_include/lan.network.user
/etc/aredn_include/wan.network.user
/etc/aredn_include/dtdlink.network.user
/etc/aredn_include/dnsmasq-user.conf
/etc/aredn_include/babel-user.conf
/etc/arednsysupgrade.d
/etc/dropbear/dropbear_dss_host_key
/etc/dropbear/dropbear_rsa_host_key
/etc/dropbear/dropbear_ecdsa_host_key
/etc/dropbear/dropbear_ed25519_host_key
/etc/dropbear/authorized_keys
/etc/local/mesh-firewall/59-custom-rules
/etc/local/uci/hsmmmesh
/etc/passwd
/etc/shadow
/etc/package_store
/etc/state/babel-state
/app/resource/css/theme.css
/app/resource/css/theme.version
```

Backing up custom files

There may be cases when you have added your own custom files on a node, and you want those files to be included in the backup archive. You can accomplish this by adding your own backup file list and placing it into the `/etc/arednsysupgrade.d/` directory on your node. For example, you could navigate to the *arednsysupgrade.d* directory and create a file called `MYCALL-wxservice-files.conf`. This will be a simple text file containing the full paths of the custom files you want to have backed up. For example:

```
# cat /etc/arednsysupgrade.d/MYCALL-wxservice-files.conf
/usr/local/wx-alert-checker.sh
/usr/local/wx-forecast-checker.sh
```

(continues on next page)

(continued from previous page)

```
/tmp/wx-alerts.txt
/www/wx-forecast.html
```

Note: The file paths listed in the `MYCALL-wxservice-files.conf` file will be included in the backup archive, and they will also be reinstalled at the time the node firmware is upgraded.

9.4.5 Restore Configuration

Once you have generated and saved a backup configuration, you can restore that previous backup to your node. This will replace the node's configuration with the settings in the backup file. Be aware that no attempt is made to validate the backup file. Also, restoring to a different type of hardware could result in unexpected behavior.

A progress bar at the bottom of the display will show the status of your download or upload. Any error messages will also be displayed in a message bar at the top of this display. You should then see a display showing that the image is being installed, along with a timer and progress indicator.

9.4.6 Advanced Firmware Options

Keep Configuration
: This is enabled by default and will allow you to retain your existing configuration settings during the firmware upgrade process. If you do not want any existing configuration settings to be retained, you can `disable` this setting and the node will come up in "firstboot" state.

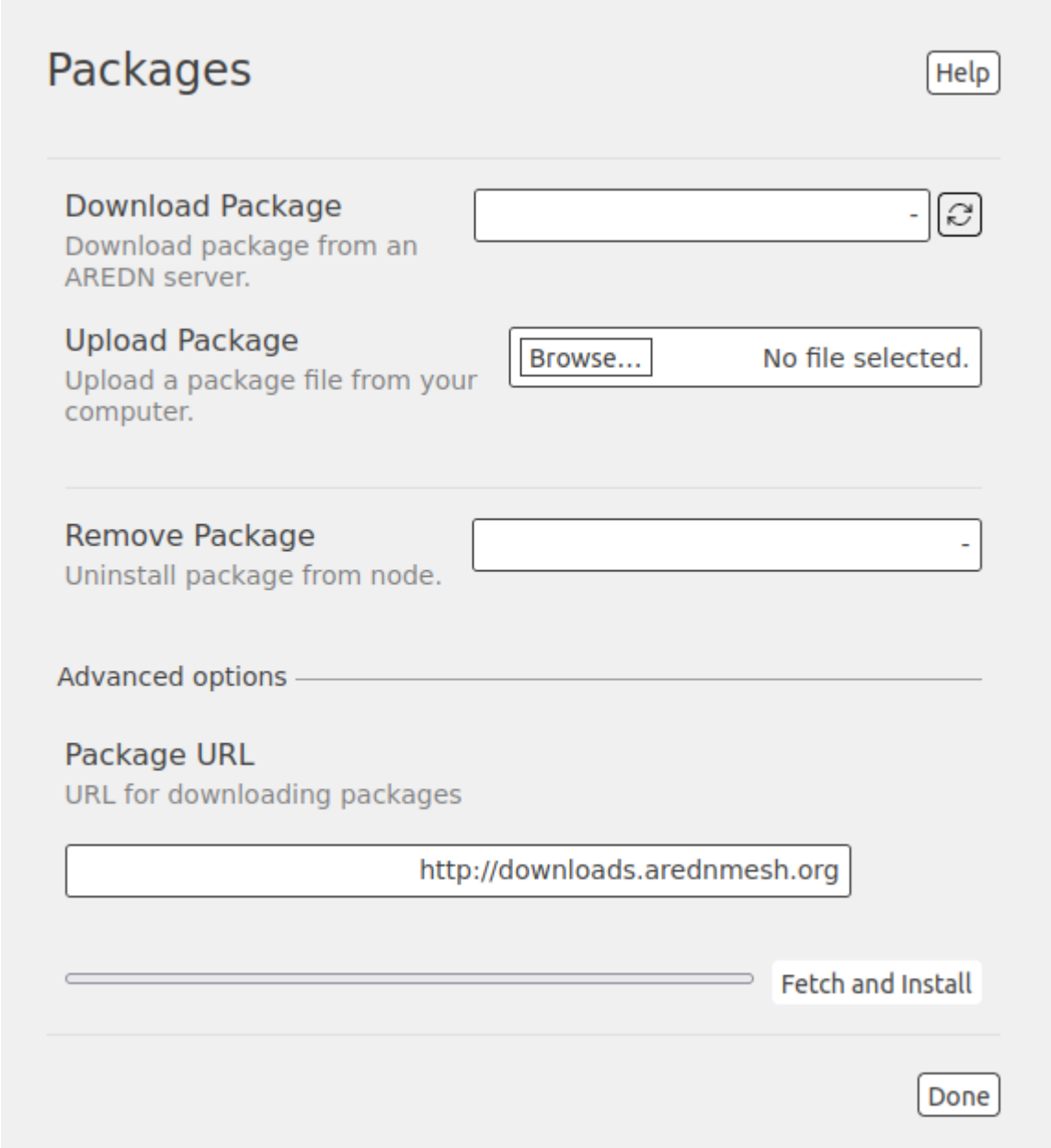
Dangerous Upgrade
: This setting allows you to disable the normal firmware compatibility safety checks that typically prevent you from loading the wrong firmware image on your node. The default setting is `disabled` which means that the safety checks remain active, and this setting should not be changed unless you have a specific reason to bypass the firmware compatibility checks. One example for using this setting would be if you mistakenly installed an incorrect firmware image and would like to correct that mistake by installing the correct firmware image.

Firmware URL
: This is the source URL that is queried by the *Download Firmware* process in order to refresh the list of available firmware for your node. The default value is `http://downloads.arednmesh.org` which allows your Internet-connected node to retrieve firmware from the AREDN® website. You can also set this firmware URL to a local server which provides firmware images.

If you are only making changes to firmware settings, you will click the `Done` button. You are then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes. However, if you are updating the node's firmware as described in the previous sections, then the `Fetch and Update` process will begin immediately and you are not required to click the *Done* button.

9.5 Package settings

Highlight and click the section displaying your node's installed package count. This display allows you to install or remove software packages on the node. When you install packages, your node will remember them in its package store. When you next upgrade your node's firmware, the package store will be retained. After the firmware upgrade your node will automatically reinstall any packages in its package store. If you originally *uploaded* the package to the node, then the package store keeps a copy of the package code itself. If you originally *downloaded* the package, then your node will attempt to re-download it. Also, if you later *remove* one of your extra packages, it will be automatically removed from the package store. Context-sensitive help is available by clicking the `Help` button.



9.5.1 Download Package

If the node has a connection to the Internet or to a local package server, it can retrieve a package from the AREDN® website or from the local server. Click the *refresh* icon at the right of the field to update the list of packages available for download. Select the package you want to install, click the `Fetch and Install` button, and wait for the package to be installed. A progress bar at the bottom of the display will show the status of the process. A status message will appear at the top of the display to indicate whether the package was installed successfully.

9.5.2 Upload Package

If you have a package file that you already downloaded to your local computer from a package repository, click the `Browse` button and navigate to the location where you saved the package file. After selecting the package, click the `Fetch and Update` button and wait for the package to be uploaded and installed. A progress bar at the bottom of the display will show the status of the upload and install. A status message will appear at the top of the display to indicate whether the package was installed successfully.

9.5.3 Remove Package

Click in the field at the right to show a list of packages currently installed on the node. Select a package and click the `Remove` button to uninstall the selected package. You will only be able to remove packages that you have added to your node. A progress bar at the bottom of the display will show the status of the remove process. A status message will appear at the top of the display to indicate whether the package was removed successfully.

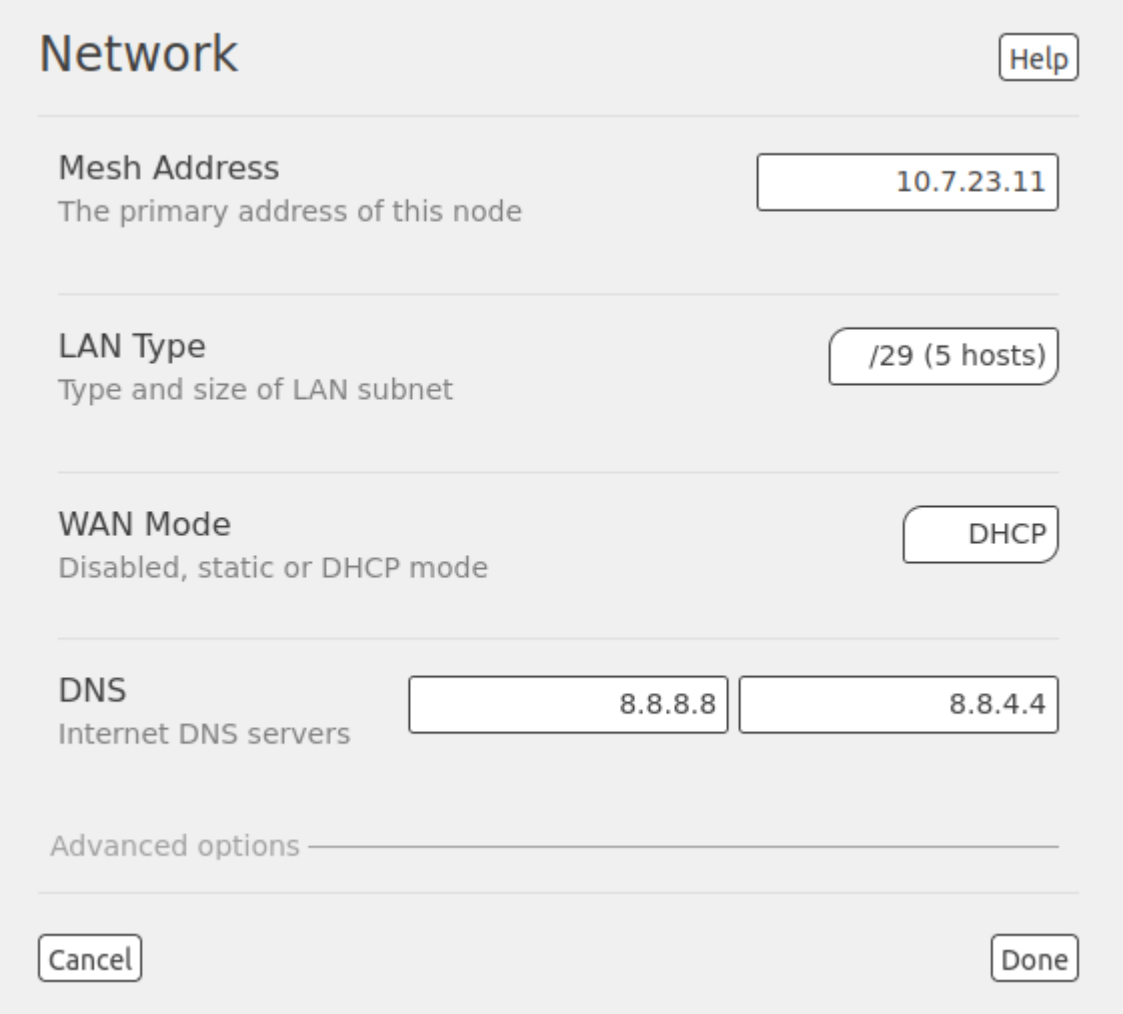
9.5.4 Advanced Package Options

Package URL
: This field contains the URL which your node will use to download packages. The default value is `http://downloads.arednmesh.org` which allows your Internet-connected node to retrieve packages from the AREDN® website. You can also set this package URL to a local server which provides packages.

If you are only making changes to package settings, you will click the `Done` button. You are then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes. However, if you are installing or removing a package as described in the previous sections, then the install or remove process will begin immediately and you are not required to click the *Done* button.

9.6 Network settings

Highlight and click the section displaying your node's network settings. This display allows you to update the network configuration on your node. Context-sensitive help is available by clicking the `Help` button.



9.6.1 Mesh address

The **Mesh Address** is the primary IP address of your node. The AREDN® firmware has been designed to simplify the process of configuring network interfaces. Network values are automatically calculated based on the unique MAC (Media Access Control) addresses on your node. Normally you will not need to change this, so keep this value unless you fully understand how the mesh works and why the defaults may not be suitable for your situation.

9.6.2 LAN

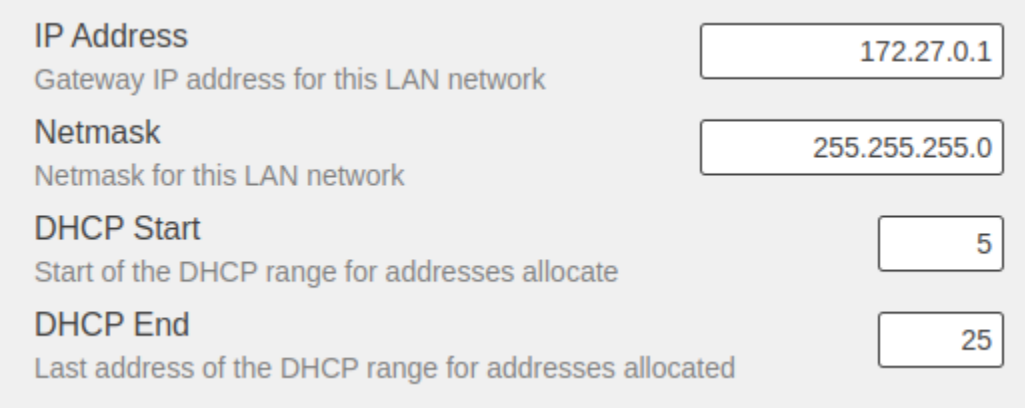
The **LAN Type** allows you to set the number of devices your node will be able to host on its Local Area Network (LAN). Click in the field at the right to see the dropdown list of options for the size of your node's LAN. The default value is `5 hosts`. It is important not to select a size that is larger than necessary because the chance of an IP address conflict on the mesh increases with the size of the subnet. The LAN subnet parameters are automatically calculated and depend on the IP address of the *Mesh* interface. If a conflict does occur it can be fixed by changing the *Mesh* IP address above.

The most common configuration is to have the LAN address space managed automatically for you. In this case the LAN shares the same address space as the mesh at large, and every host on the LAN has direct access to and from the mesh. You have the option of selecting the size of the LAN subnet which can accommodate either 1, 5, 13, or 29 LAN hosts. A single host subnet can be useful for either a single server or a separate network router using its own NAT which is capable of more advanced routing functions than those available on a mesh node. This design minimizes the amount of manual effort needed to provide services to the mesh, since many services do not work well if they are hosted behind a NAT (Network Address Translation) router.

When you connect a device to your node's LAN, not only will it have an IP address in the LAN IP address range, but it is best practice for LAN device to obtain its DNS Server information automatically from the node. Be aware that if a LAN device does not use the DNS Server entry provided by the node to which it is connected, then that device will be unable to resolve hostnames on the mesh network. Also, hard-coding a device's DNS Server entry with the mesh node's IP address could result in unexpected failures if that IP address changes.

LAN NAT Mode

Another choice for *LAN Type* is `NAT` and in this mode the LAN is isolated from the mesh. All outgoing traffic has its source address modified to be the *Mesh* IP address of the node itself. This is the same way that most home routers use an Internet connection, and all services provided by computers on the LAN can only be accessed from the mesh using port forwarding rules.

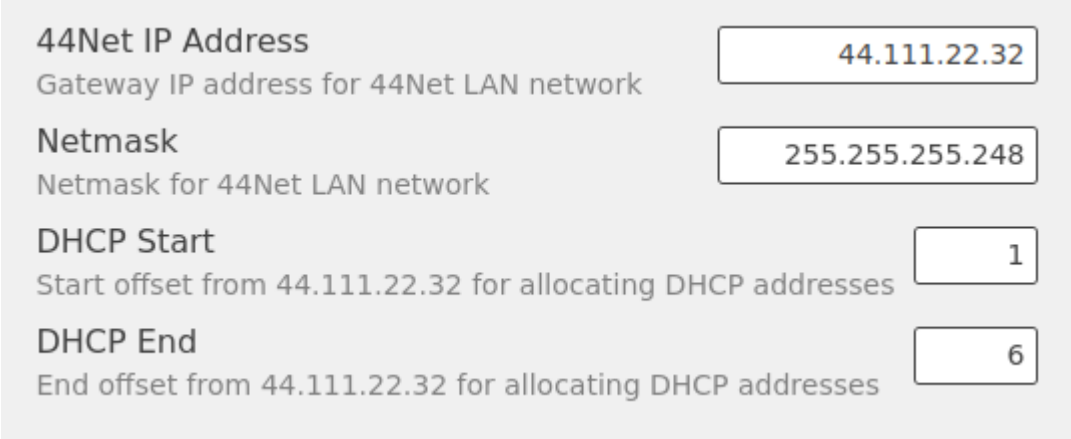


In `NAT` mode you are responsible for managing the IP address space of your node's LAN network.

Enter the LAN IP address and netmask in dotted decimal format. Specify the final octet of the IP address that your node's DHCP service will use as its *DHCP Start* address as well as the *DHCP End* address, which defines the IP address range that will be provided via DHCP for LAN devices.

44Net Mode

Another choice for *LAN Type* is `44Net` and this mode allows you to use IP addresses from the AMPRnet address space (44.0.0.0/9 to 44.128.0.0/10).



Enter the **44Net IP Address** and **Netmask** in dotted decimal format. Specify the offset of the IP address that your node's DHCP service will use as its *DHCP Start* address as well as the offset for the *DHCP End* address, which defines the IP address range that will be provided via DHCP for LAN devices.

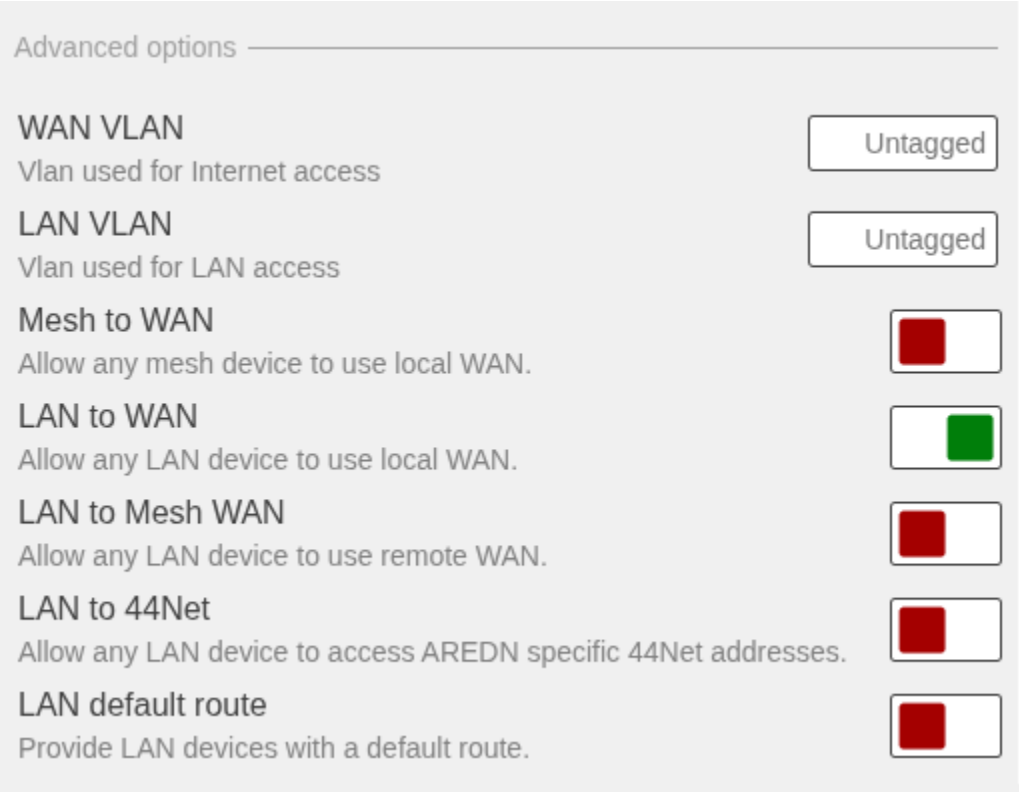
9.6.3 WAN

WAN Mode
: This specifies whether your node's WAN interface is enabled, and if so, how it gets its IP address. The default is to use DHCP, so the WAN IP address is assigned to your node by your Internet router. If you select `Static` you will see several new fields which allow you to specify the IP address, netmask in dotted decimal format, and gateway IP address.

DNS
: These two fields allow you to enter the IP addresses of the DNS (Domain Name System) servers of your choice. By default Google's DNS servers are listed because their name resolution servers are configured to detect error conditions properly and to report them correctly.

9.6.4 Advanced network options

Additional options will be displayed when you click **Advanced Options**.



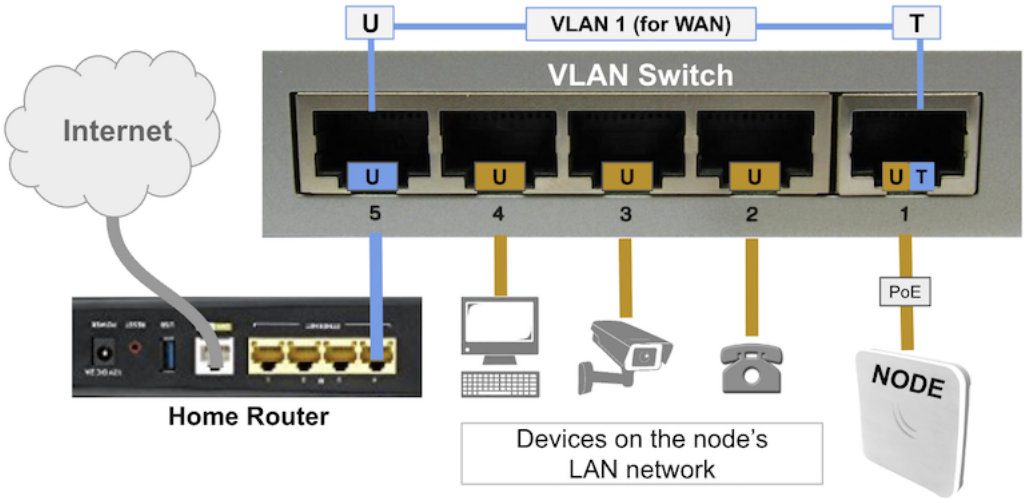
WAN & LAN VLANs for single port nodes

Many of the devices used as AREDN® nodes have only one Ethernet port, and several types of network traffic must share that single port. The AREDN® firmware implements VLANs (Virtual Local Area Network) in order to accomplish this. Different types of traffic are tagged to identify the network to which they belong.

By default the WAN uses `VLAN 1` and the LAN is `Untagged` on single port devices. These can be changed if your network requires something different. Enter the VLAN number or leave the field blank for *Untagged*. If you change this setting and want to use a single digit identifier, use numbers 5 but do not use any number larger than can be supported by your network equipment. Different types of network equipment can support various numbers of VLANS, but the maximum number is limited by the 802.1Q standard to no more than 4094.

The following VLANs are preconfigured in the AREDN® firmware:

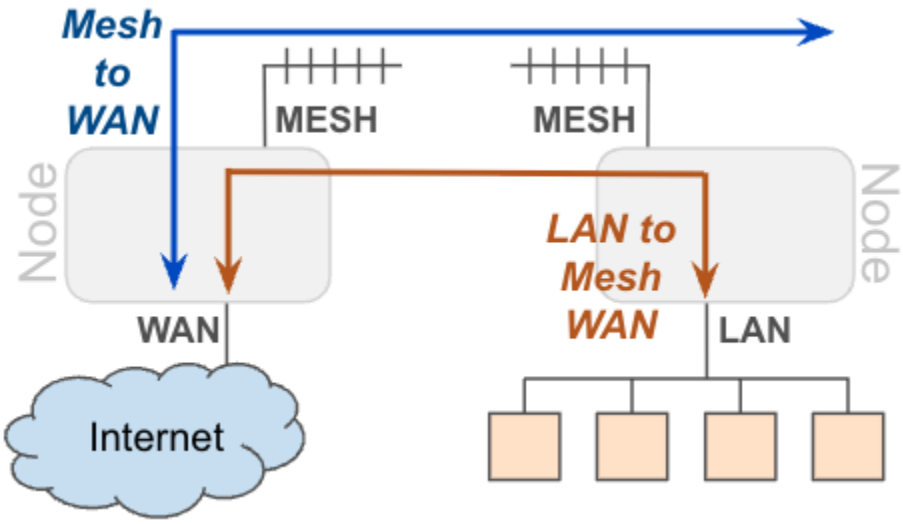
- Untagged identifies LAN traffic from devices on the local area network
- VLAN 1 identifies WAN traffic to your node from the Internet or another external network
- VLAN 2 identifies traffic from a DTD node directly connected to your node



It is important to understand AREDN® VLANs when configuring network smart switches for single-port nodes to access the Internet, tunneling, or DtD linking of nodes. There are some useful tutorials available on the AREDN® website for configuring VLAN-capable switches: Video or Text+Images. You can get the latest information about the specific port configured as the node's WAN port from the AREDN® website here: Ethernet Port Usage.

Mesh to WAN

Enabling this switch will allow your node to route traffic from its Mesh interface to/from its WAN interface. This allows any device on the mesh network to use the WAN on your node, typically for accessing the Internet. It is usually not desirable to route Internet traffic over your Mesh interface. AREDN® is an FCC Part 97 amateur radio network, so be sure that any traffic which will be sent over the radio complies with FCC Part 97 rules. If you want local devices to have wireless Internet access, consider using an FCC Part 15 access point instead of your node's WAN gateway. The default value is `disabled` and it is recommended that you keep this default unless there is a special reason to enable it.

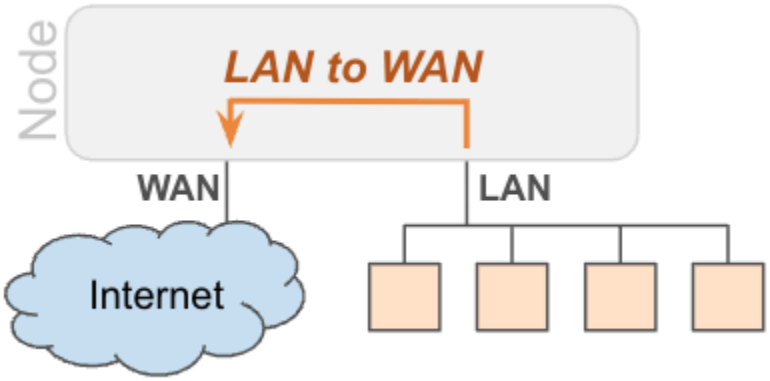


LAN to Mesh WAN

There may be times when your node has its own LAN devices, but your node does not provide WAN Internet access for them. Enabling this option will allow your node's LAN devices to find and use an Internet connection that might be available from another node across the mesh network. This option is `disabled` by default.

LAN to WAN

The default value is `enabled` which allows devices on your node's LAN to access your node's WAN network. Setting this value to `disabled` will prevent LAN devices from accessing the WAN, which means that your LAN hosts will not be able to reach the Internet even if your node has Internet access via its WAN. You may need to disable WAN access if your device needs to be connected to two networks at once, such as an Ethernet connection to your node as well as a wifi connection to a local served agency network.



LAN to 44Net

The default value is `enabled` which provides a 44Net route for any LAN device on your node, even if your default route is disabled.

LAN default route

Your node's DHCP server will provide routes to its LAN devices so they can access any available networks. A default route is required for WAN access, and that is provided automatically if **LAN to WAN** is *enabled* as discussed above. However, some LAN devices (such as certain IP cameras) may not support DHCP option 121, so they will require a default route in order to access the mesh network. Setting this value to `enabled` will provide a default route to those devices. If a LAN device is connected to two networks at once, such as an Ethernet connection to your node as well as a wifi connection to a local served agency network, care should be taken to understand how the device will deal with default routes for more than one network. The default value is `disabled` and you should not enable it unless you have a special reason to do so.

Custom firewall rules

There may be cases when you want to create additional firewall rules to allow specific traffic through your node. You can define custom firewall rules by entering them into the `/etc/config.mesh/firewall.user` file on your node. This feature is for advanced users and assumes that you have the skills to construct *nftables* firewall rule statements. The example below is for a node that has its **NTP Server** switch enabled, which allows only LAN-connected devices to use your node as an NTP server. If you also want to allow local DTD-linked devices to use your node's NTP server, you could add a custom firewall rule as shown below.

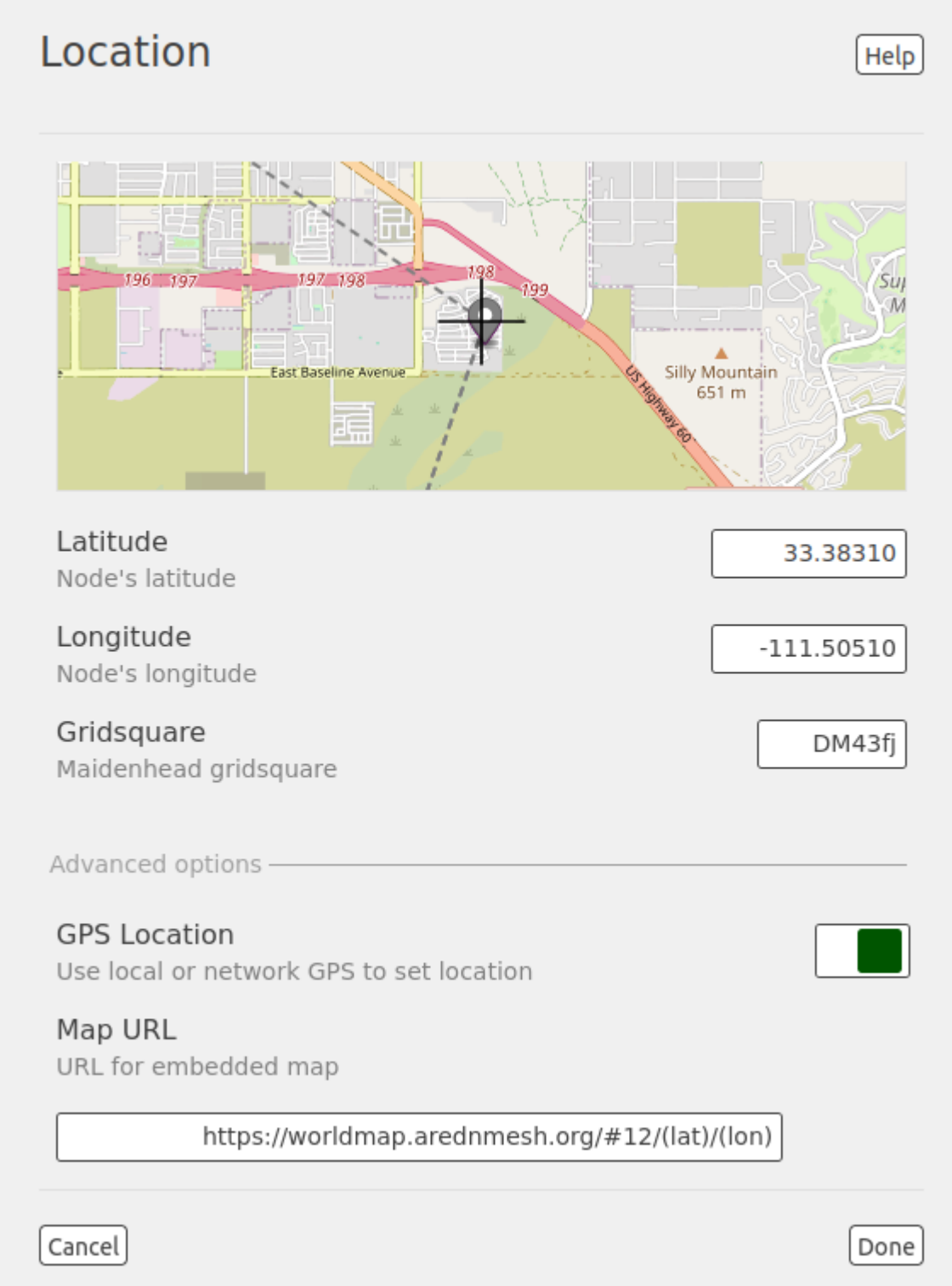
```
# This file is interpreted as shell script.
# Put your custom nft rules here, they will
# be executed with each firewall (re-)start.
nft insert rule inet fw4 input_dtdlink udp dport 123 accept
```

After creating custom rules, you will need to reboot your node (or restart the node's firewall) for the rules to become active. The contents of `firewall.user` will be included automatically in the backup file when you perform a **Backup** of your node's configuration.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

9.7 Location settings

Highlight and click the section displaying your node's location. This display allows you to update the location settings on your node. Context-sensitive help is available by clicking the `Help` button.



Any values you enter should be in decimal format, and the values in these three fields are linked. Any changes made will automatically update the fields and the map thumbnail. You can also change the location information by clicking on the map and panning around to set your location. As you pan the map, the location values will follow your movements automatically.

Location information is used to determine the distance between your node and others, and it is required for optimizing connection latency and bandwidth. A Maidenhead grid square is a six character designation of a node's location. A grid square identifier consists of two uppercase letters, two digits, and two lowercase letters. Each grid square is approximately 3x4 miles in size.

Additional options will be displayed when you click **Advanced Options**.

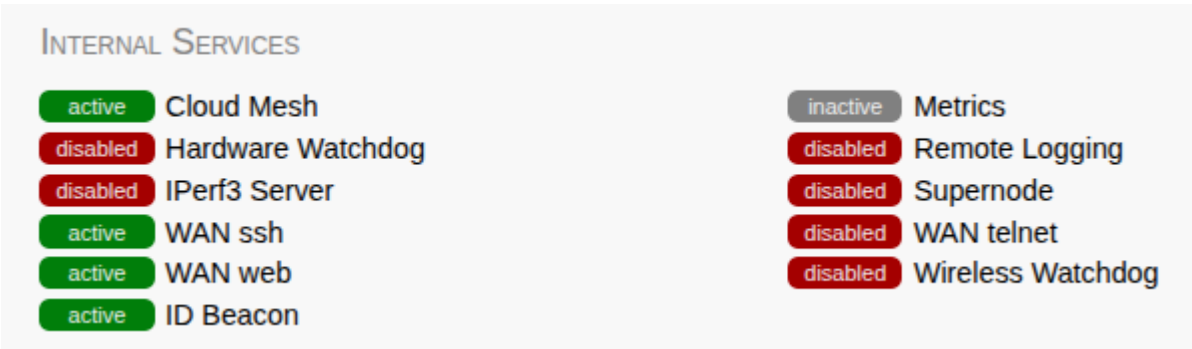
By default your node will attempt to set its location from a local GPS source. If you want to disable this behavior, slide the **GPS Location** switch to the *off* position.

The **Map URL** is used to embed maps in your node's displays. The default value is `https://worldmap.arednmesh.org/#12/(lat)/(lon)` which attempts to get map data from the AREDN® server. The (lat) and (lon) parameters in the URL are substitutes with your GPS coordinates before the map is rendered. If there is a local map tile server available on your mesh network, then you can point your node to the local server for its map data.

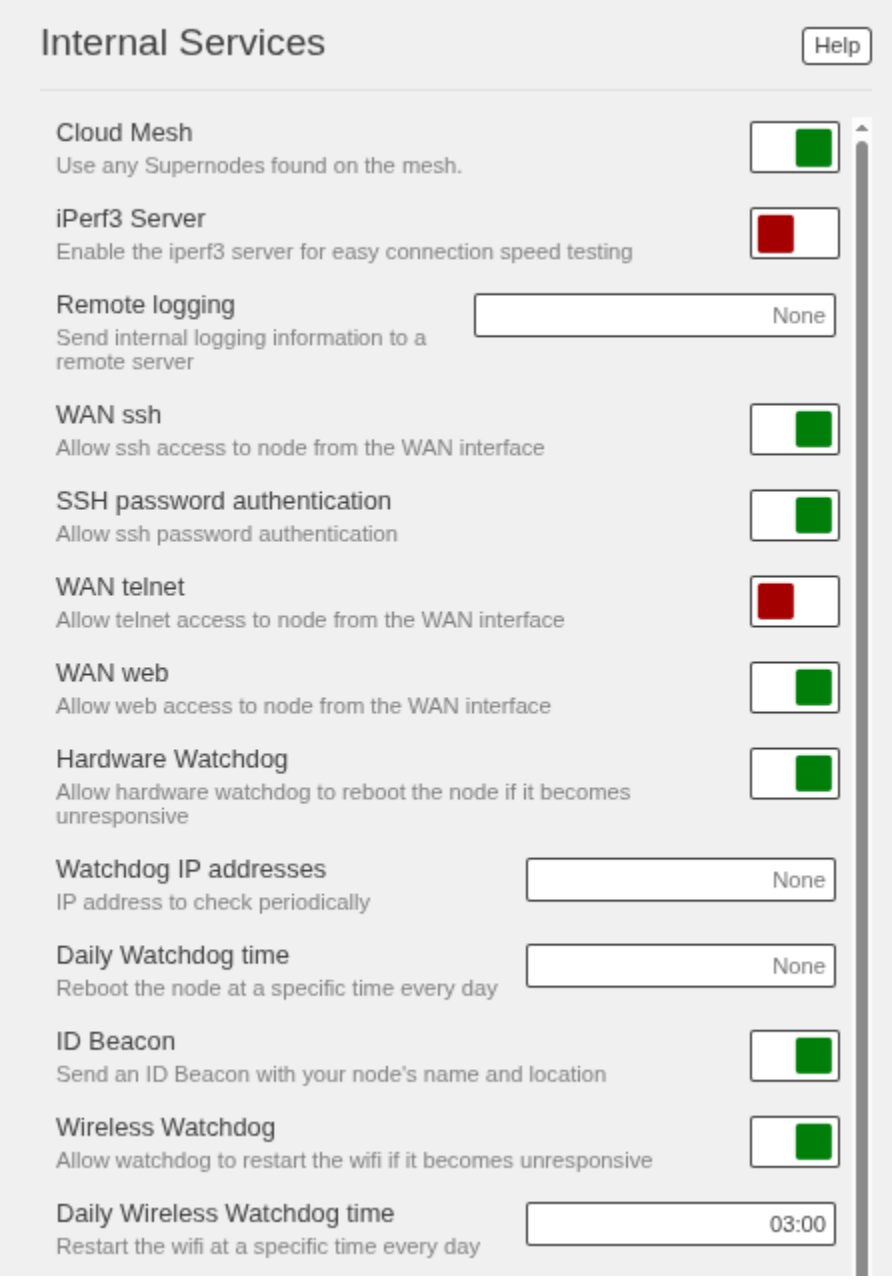
You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

9.8 Internal Services

When you are logged in as *admin* you will see an Internal Services status display at the top of the center column. This shows the state of each of the listed services, which will be described below in more detail. The `Metrics` status is informational only and simply indicates whether this node is currently being monitored by providing metrics to an external service (such as Prometheus). The `Supernode` status is informational and will only appear on nodes capable of being a Supernode, indicating whether this node is currently configured as a Supernode.



Highlight and click the section displaying your node's **Internal Services**, which allows you to manage the internal settings on your node. Context-sensitive help is available by clicking the `Help` button.



Cloud Mesh

This switch allows your node to use any available Supernode on your local mesh. Supernodes are a way to link multiple mesh island networks in a safe and efficient way. If your local node is part of a network with a Supernode then you have the ability to view other nodes which are part of the Cloud Mesh network. This feature is `enabled` by default. Clicking the Cloud Mesh icon will navigate to the mesh status display of the closest Supernode available to your device. For further information see the *Supernode Architecture* description in the **Network**

Topologies section of the **Network Design Guide**.

 You may connect to any node on the worldwide mesh by clicking your node's Cloud Mesh icon on the left nav bar. Disable this option if you never want your node to provide a method of accessing devices on the worldwide mesh network.

iPerf3 Server
This switch enables the built-in iperf3 tools on your node. This makes it easy to perform throughput tests between nodes in the network. The client and server are only invoked on demand, so there is no performance impact on the node except during testing. The default value is `enabled`. If you do not want your node to participate in any remote iperf3 tests then you can `disable` its ability to respond to those queries.

Remote Logging
The limited amount of memory for local node logs means that older information will roll off, and all log history is lost when your node is rebooted. By entering the URI for a remote log server, you can send your node's log info to a server using the syslog protocol. The format for this option is `udp://ip-address:port` or `tcp://ip-address:port`. Leave this field blank if no remote logging is desired.

WAN ssh
This switch enables SSH access to your node on its WAN interface. Disabling this option will not prevent SSH access to your node from the Mesh and LAN interfaces.

SSH password authentication
This switch allows *ssh* password authentication, which is `enabled` by default. Disabling this option forces the use of previously uploaded ssh keys.

WAN telnet
This switch enables *telnet* access to your node on its WAN interface. Disabling this option will not prevent *telnet* access to your node from the Mesh and LAN interfaces.

WAN web
This switch enables http/https access to your node on its WAN interface. Disabling this option will not prevent http/https access to your node from the Mesh and LAN interfaces.

Hardware Watchdog
Hardware watchdog is a background monitor that keeps track of core node processes. If any of the processes has issues, it will reboot the node. This feature is `disabled` by default. Currently the set of node processes that are monitored include dnsmasq, telnetd, dropbear, uhttpd, and babeld. Hardware watchdog events are logged in the standard log on the node. Because the watchdog operates at the hardware level, the node will still reboot itself even if the kernel crashes.

> **Attention:** Be aware that you must disable Hardware Watchdog and reboot your node before you can upgrade the firmware, since Hardware Watchdog may interfere with the

normal upgrade process.

If Hardware Watchdog is `enabled` the following fields will also be displayed, otherwise they will be hidden.

Watchdog IP Addresses
: You may include one or more IP addresses, at least one of which should always be pingable. Watchdog will be reboot the node if none of the IP addresses is reachable across the network. Enter IP addresses as a whitespace-delimited list. It is strongly recommended that you keep this list to an absolute minimum. Too many addresses can take a long time to check, especially if several are unavailable. This can result in reboots if the testing is not performed before the watchdog timer expires. Ideally use only one address.

Daily Watchdog Time
: This field allows you to set a specific time of the day (between 00:00 and 23:59) to restart the node automatically. The node must get its time from NTP or GPS in order for this reboot to occur.

ID Beacon
: This switch is `enabled` by default, which tells your node to send a beacon that contains your Amateur Radio callsign (as well as the node's location) every few minutes. Periodic Amateur Radio station identification is a requirement in many regions in order to comply with local regulations.

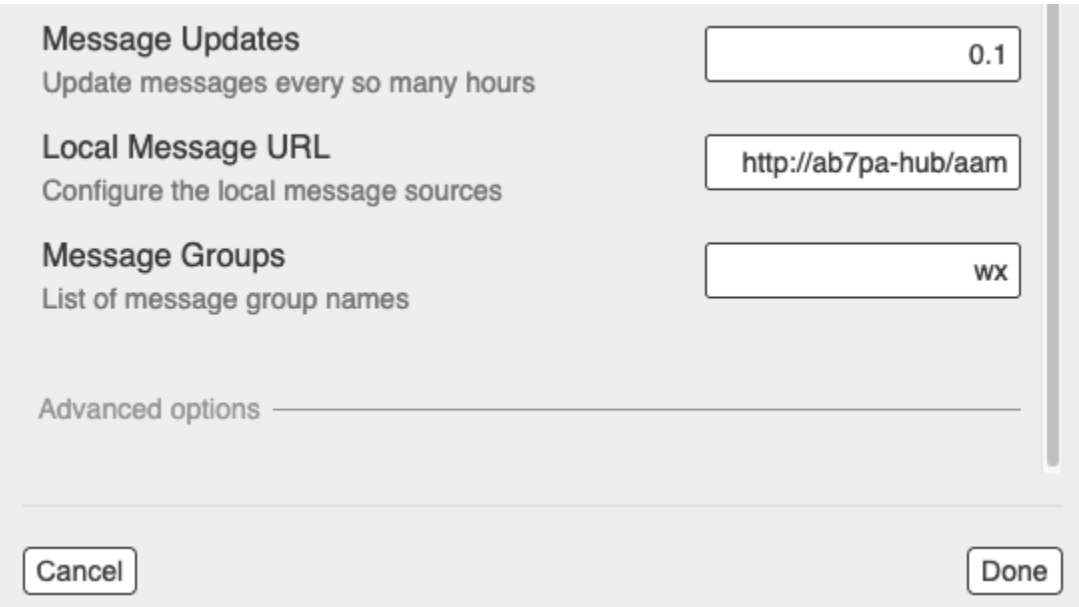
Wireless Watchdog
: This background monitor will restart the mesh radio if it becomes unresponsive. If Wireless Watchdog is `enabled` the following field will also be displayed, otherwise it will be hidden.

Daily Wireless Watchdog Time
: This field allows you to set a specific time of the day (between 00:00 and 23:59) to restart the radio automatically.

PoE and USB Power Passthrough
: These settings will only appear if you have node hardware which supports PoE or USB power passthrough. One example is the *Mikrotik hAP ac lite* which provides one USB-A power jack (~5v) as well as PoE power passthrough on Ethernet port 5 (~22v). You are allowed to enable or disable power passthrough on nodes with ports that support this feature.

9.8.1 AREDN® Alert Messages

AREDN® Alert Messages are displayed on the status page of nodes. The AREDN® development team can post messages which Internet-connected nodes download and display, or you can implement your own local message server from which your nodes can pull their messages.



Message Updates
: Enter a number in this field which represents the number of hours you want your node to wait before pulling its messages. Decimal fractions of an hour are allowed (for example, `0.5` for every 30 minutes). The default value is 1 hour between updates.

Local Message URL
: This field allows you to enter the URL for a local message source. If you configure a local message server, then even nodes without Internet access can receive alert messages via RF or other local links. The message source consists of a mesh-connected web server which allows nodes to query the URL you enter here. You can consult your local mesh web server administrator to obtain the correct URL for the local message repository. Enter the URL without a trailing backslash.

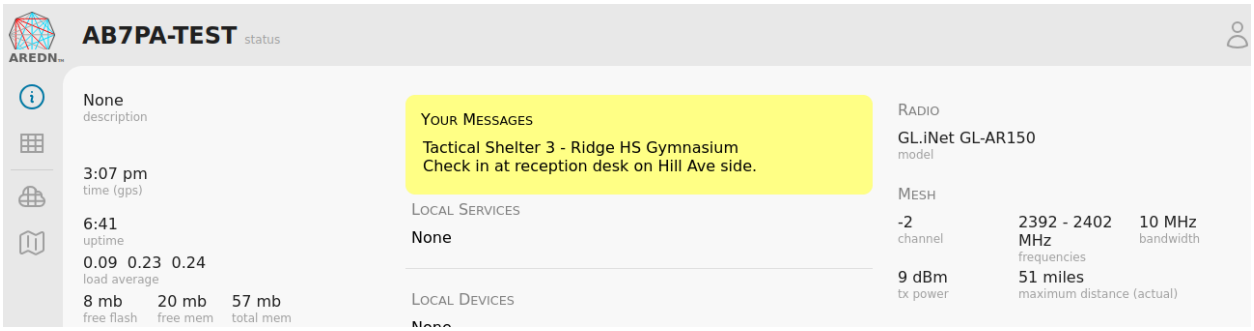
Message Groups
: In addition to local messages addressed by name to specific nodes, it is possible to subscribe to group messages. Multiple group names can be added to this field as a comma delimited list. Group messages are retrieved from the web server specified in the *Local Message URL* field. The following are example grouping ideas:

  - Geographic regions (county or neighborhood)
  - Weather alerts
  - Wildfire, flooding, or other emergency alerts
  - SKYWARN activations, DHS threat level

The following file naming convention should be used for the web server's message repository:

- Create text files for individual node messages by using only lowercase characters with the exact node name followed by `.txt`. Whitespace characters are not allowed in node names, and do not append `.local.mesh` to the node name. An example node-specific message might be contained in `ab7pa-test.txt`.
- Create text files for group messages by using only lowercase characters with the group name followed by `.txt` extension. Whitespace characters are not allowed in group names. An example group message for the current weather conditions (group name = wx) might be contained in `wx.txt`.
- To create a broadcast message intended for all local nodes, enter your message text in a file named `all.txt` using only lowercase characters for the filename.

Alert messages are displayed in a highlighted text box at the top of the node status page.



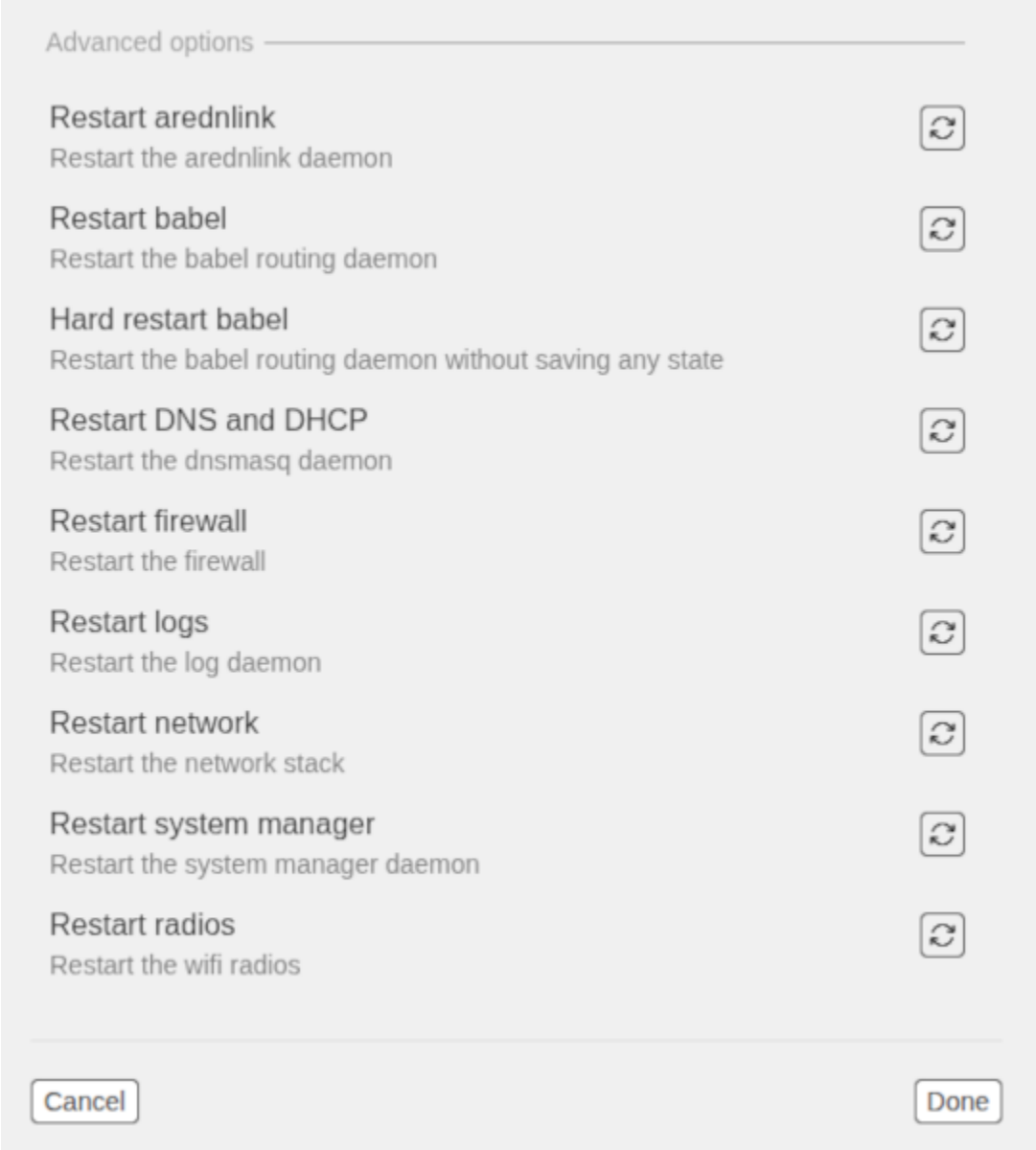
9.8.2 Advanced Internal Service Options

Additional options will be displayed when you click **Advanced Options**.

Restart firmware processes

Specific firmware processes can also be restarted without having to perform a full reboot of your node.

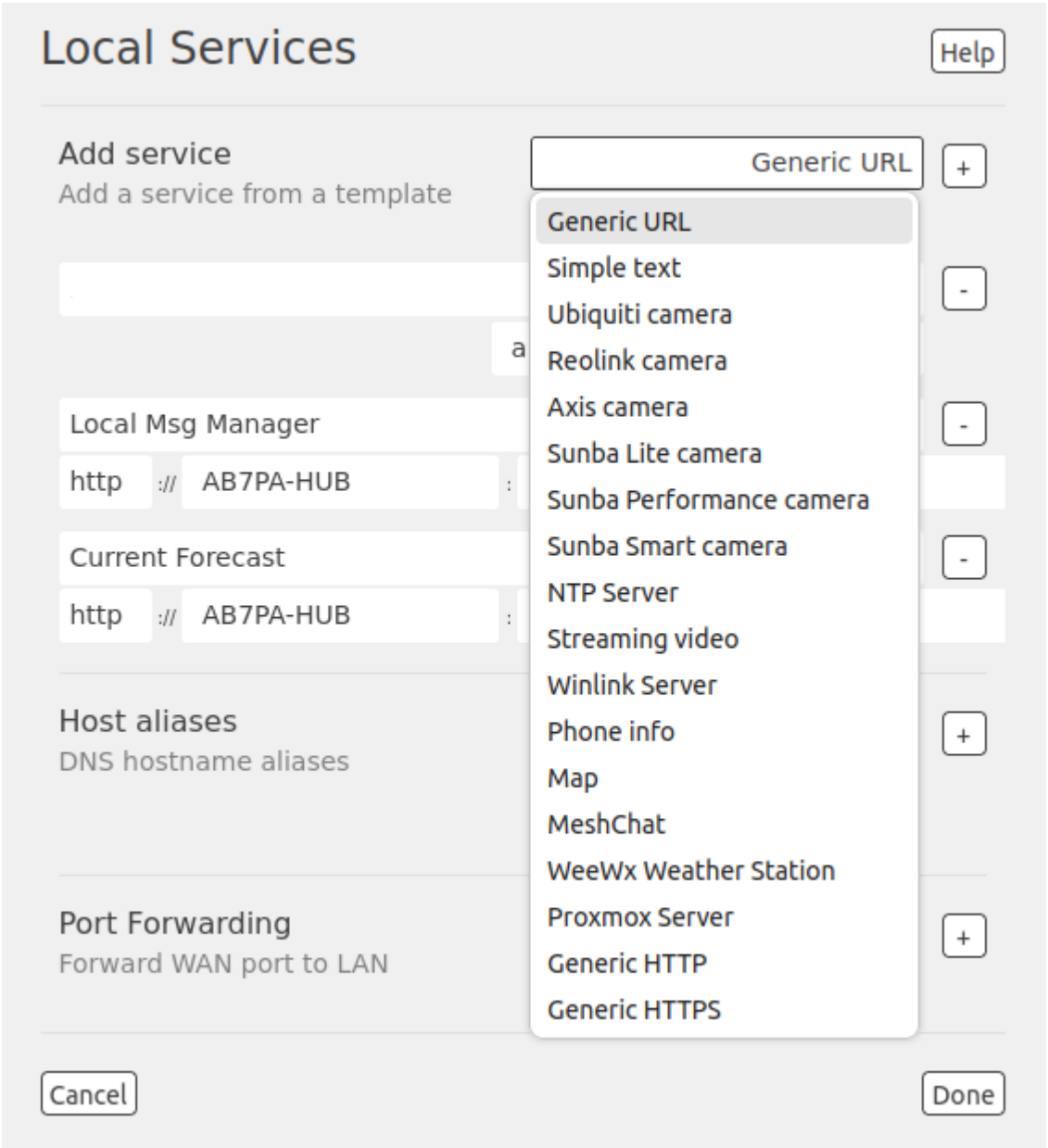

Click the icon at the right side of any service to restart that service. The icon will spin briefly to indicate that the process has been restarted. Note that restarting some of the internal services may disconnect your node from the network, and it may require some time for the connections to be reestablished.



After reviewing the **Internal Services** display, you can click the `Cancel` button to ignore any changes you made. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

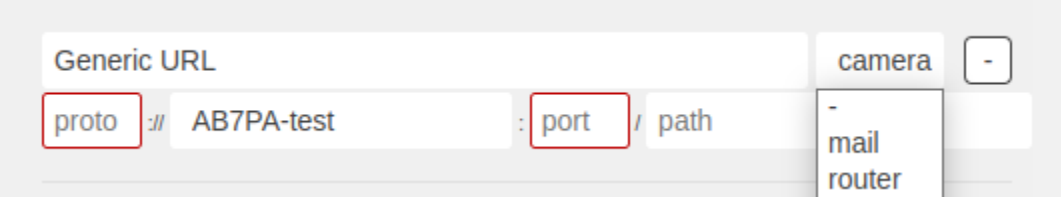
9.9 Local Services

Highlight and click the section displaying your node's local services. The **Local Services** display allows you to manage the services which will be available on your node. The purpose of the network is to transport data for the services which are being used. Network services may include keyboard-to-keyboard chat or email programs, document sharing applications, Voice over IP phone or video conferencing services, streaming video from surveillance cameras, and a variety of other network-enabled features. Services can run on the node itself or on any of its LAN-connected devices. Context-sensitive help is available by clicking the `Help` button.



9.9.1 Adding a Service

To add a service, click in the field to the right and select the type of service you want to add. Then click the [+] icon to add a row to your services list for the new service of the selected type. You will provide different parameters for the new entry based on the type of service selected.



Generic URL service template

This template allows you to enter a descriptive *service name* to clearly identify your service ("Generic URL" is a placeholder). Click in the field to the right of the *service name* to select from the dropdown list the type of icon that will be displayed for this service (if any). The icon you choose will be displayed to the right of the service name on **mesh status** pages.

Icon List by Name

Icon	Name	Icon	Name	Icon	Name
	battery		map		solar
	camera		phone		syslog
	chart		power		time
	chat		radio		video
	Internet		router or switch		weather
	mail or winlink		server		wiki

In the *protocol* field on the next row, enter the protocol to use for this service. Common protocols include `http` for website services and `ftp` for file transfer services. Other services may use other protocols. From the dropdown list in the next field, select the node or host on which this service is running. If you defined *Host Aliases* (described below), you will see these host aliases in the dropdown list.

In the next field enter the network port on which the host is listening for service connections. There may be several applications provided through a single web server on a node or host using a single port, and in that case a valid application *Path* must be entered after the port number. In other cases the network port alone will uniquely identify the application or program that is listening for user connections to that service. You can find additional information on ports at the following link: Network Ports.

Simple text service template

This template allows you to create an informational label which is not clickable. Enter a descriptive label ("Simple text" is a placeholder). Click in the field to the right of the text label to select from the dropdown list the type of icon that will be displayed for this label (if any). The icon you choose will be displayed to the right of the service name on **mesh status** pages. From the dropdown list in the next field, select the node or host with which this label is associated. If you defined *Host Aliases* (described below), you will see these host aliases in the dropdown list.

Network time service template
To advertise a local NTP server, select the *NTP Server* template. The required field values are all filled for you. You can change any of the defaults that are not appropriate for your situation.

Additional service templates
Additional templates have been created for common services, with the goal of making it easier to define these services on your nodes. These templates fill in some of the fields with typical values, while allowing you to customize the information appropriately. Templates exist for several types of IP cameras as well as Winlink, MeshChat, WeeWx, Mapping, Proxmox, and web-based services.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

9.9.2 Viewing, Editing, and Deleting Services

On the **Local Services** display your services are listed as a series of rows. You can change any of the fields for any of the services in this list. If you want to delete a service row, click the [-] icon on the right side of that row.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

Service Advertisement Process

The routing protocol will propagate service entries to other nodes across the network. Once every hour your node will verify that their own service entries are valid. Your node will **not** propagate services across the network if it finds any of these conditions after three attempts:

1. The LAN host is not pingable from your node
2. There is no service listening on the specified port
3. An HTTP service does not return a *success* status code

The node's *Advertised Services* list will still show the defined service (with an alert icon and hover text marking it as non-advertised), but your node will not actually *advertise* that service to the

network. If the service URL becomes reachable in the future or if the dependent package is later installed, then your node will resume advertising the service across the network.

9.9.3 Managing Host Aliases

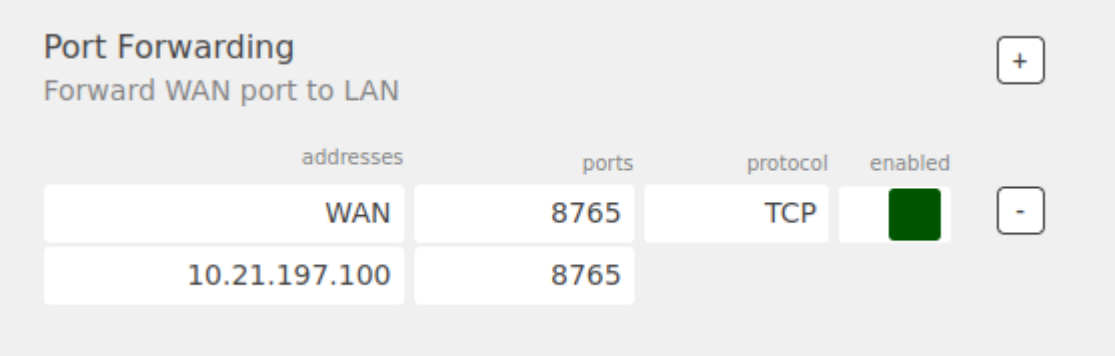
Host Aliases provide a way for you to create a hostname alias for a device on your node's LAN. This can be useful if you want a computer or device on your LAN to be identified by something other than its actual hostname. Your Host Alias will be propagated across the network even if the actual hostname has *Do Not Propagate* checked in its DHCP Reservation, allowing you to hide the actual hostname while still advertising the alias on the mesh. Once an alias is defined, it will become available for creating local services (described above).

To create an alias, click the [+] icon on the right and enter an alias name in the first field. The alias should be prefixed with your callsign in order to follow the naming convention used when defining any unique host on the network. Then use the dropdown selector to choose the name or IP Address of the existing host for which you are defining the alias. Once you have entered these values, you can change any of the fields in any of the aliases. If you want to delete an alias, click the [-] icon on the right side of that row.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

9.9.4 Port Forwarding

There may be situations where your node must act as an intermediary, typically between a remote client device and a server device on your node's LAN network. More information can be found at this link for Port Forwarding.



To create a port forwarding rule, click the [+] icon on the right. Unless the LAN is in NAT mode, port forwarding is only meaningful for WAN-connected nodes so you will only be allowed to create rules for the WAN interface. If in NAT mode you may select the WAN, Mesh, or both Mesh & WAN interfaces when defining your port forwarding rule.

For inbound port, enter a single port number or a range of ports separated by the dash character. Click in the *protocol* field to select TCP, UDP, or both. Use the switch on the right to enable or disable this port forwarding rule. On the next row, click in the IP address / hostname field to select from the dropdown list a LAN host to process the requests. In the next field, enter the *port* or the first port in the range on which that host is listening for those requests.

To delete a port forwarding rule, click the [-] icon on the right of the existing row for the rule you wish to delete. You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes.

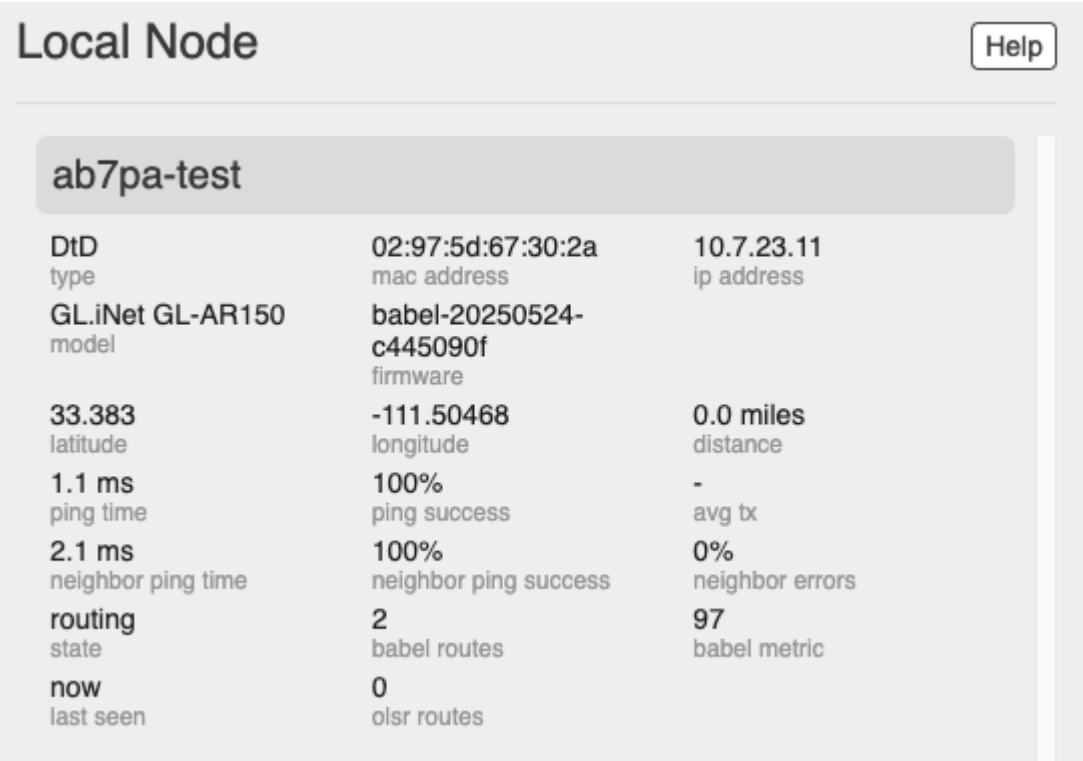
9.10 Local Devices

This section displays any devices that are directly connected to your node's LAN network. There is no *admin* action available.

9.11 Local Nodes

As described in **Node Status**, this section shows any local DTD nodes that are directly connected to your node. In order to be considered "local" the GPS coordinates entered in the *Location* section must be within 100 meters of the local neighbor. Context-sensitive help is available by clicking the `Help` button.

The node name of each Local Node is a clickable link which will navigate to that node's status page. When you hover over the row of any Local Node, a gray background appears which indicates that row is selected. If you click in the selected row (but not directly on the node name link), the **Local Node** popup will be displayed which provides more detailed information about your node's connection to the selected local node.



The following details may be displayed if available for this node's connection to your node, from top to bottom & left to right:

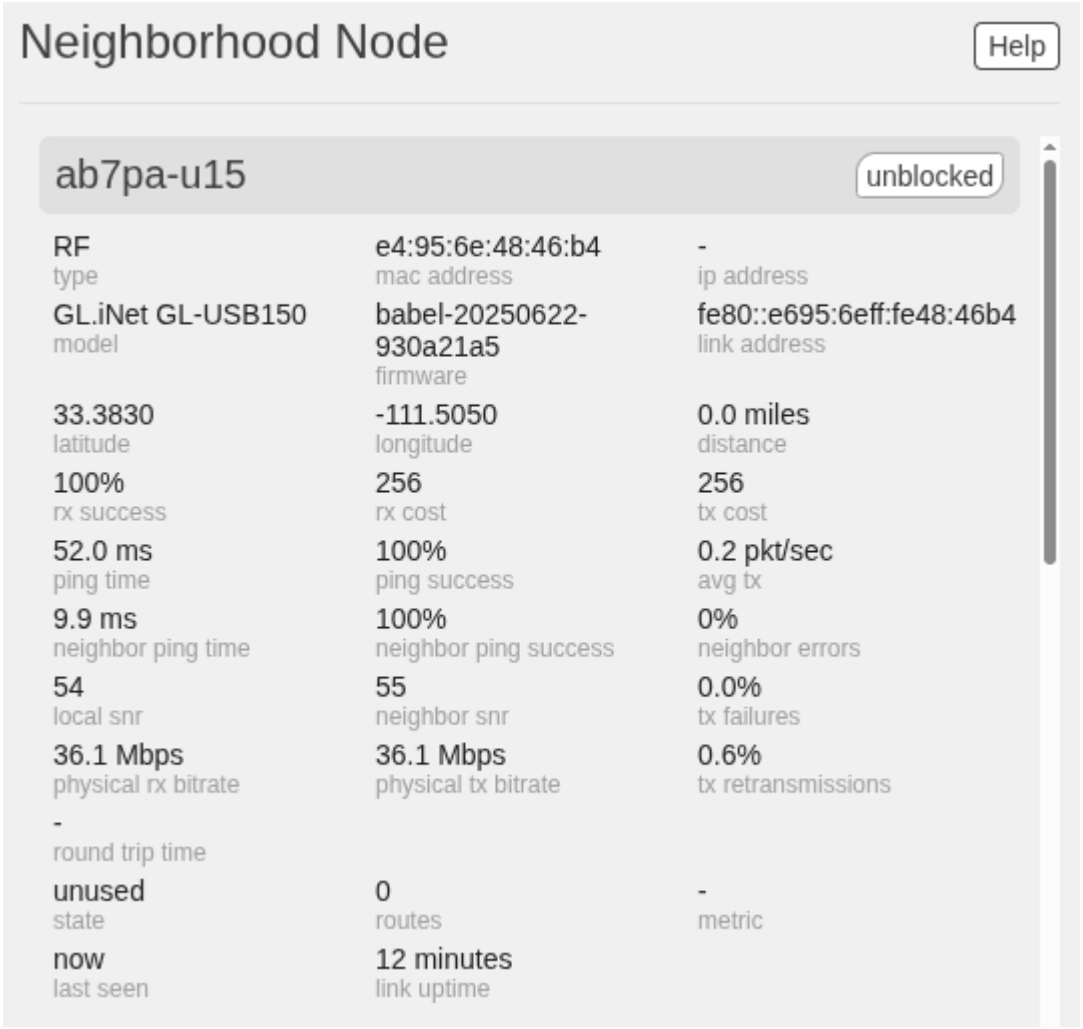
- TYPE (DTD), mac address, and ip address
- model, firmware version, and link address (Babel uses IPv6)
- latitude, longitude, and distance
- rx success rate, rx cost, tx cost
- ping time, ping success rate, and average packets per second
- neighbor ping time, ping success rate, and errors

- link state, number of Babel routes, Babel metric
- last seen, link uptime

9.12 Neighborhood Nodes

As described in the **Node Status** section, this area shows a list of neighbor devices that are directly connected to your node. Context-sensitive help is available by clicking the `Help` button.

The node name of each Neighborhood Node is a clickable link which will navigate to that node's status page. When you hover over the row of any Neighborhood Node, a gray background appears which indicates that row is selected. If you click in the selected row (but not directly on the node name link), the **Neighborhood Node** popup will be displayed which provides more detailed information about your node's connection to the selected local node. This provides an excellent troubleshooting tool for diagnosing issues with node connections, especially via RF.

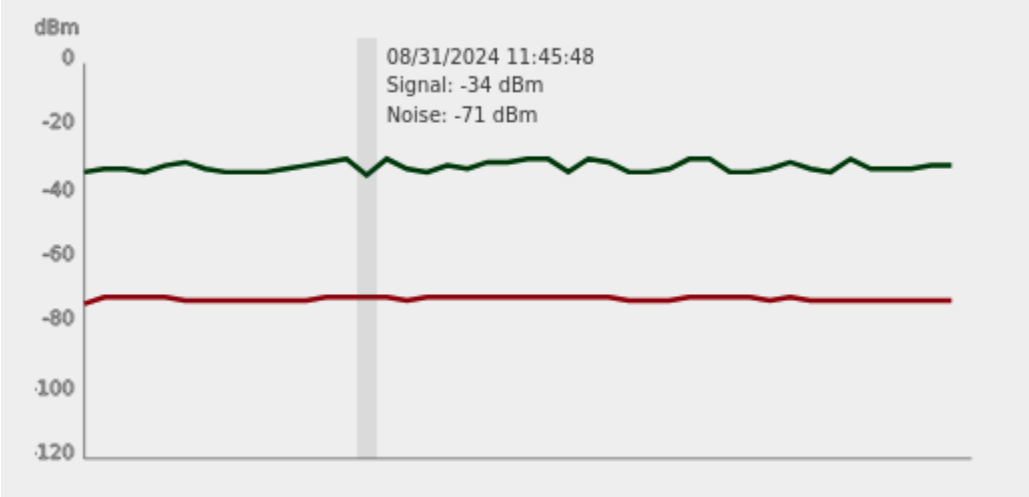


For nodes having multiple RF connections, there will be a field that shows the current link status to

the right of the node name. Clicking in this field will give you options for handling the RF link to this node, including the ability to block or unblock that node's traffic. You are not allowed to block a link that is your node's only connection to the network.

The following details may be displayed if available for this node's connection to your node, from top to bottom & left to right:

- TYPE, mac address, and ip address
- model, firmware version, and link address (IPv6 for Babel)
- latitude, longitude, and distance
- rx success rate, rx cost, tx cost
- ping time, ping success rate, and average packets per second
- neighbor ping time, ping success rate, errors
- local SNR, neighbor snr, and transmit failure rate
- physical receive bitrate, physical transmit bitrate, and retransmissions
- link state, active routes, Babel metric
- last seen, link uptime



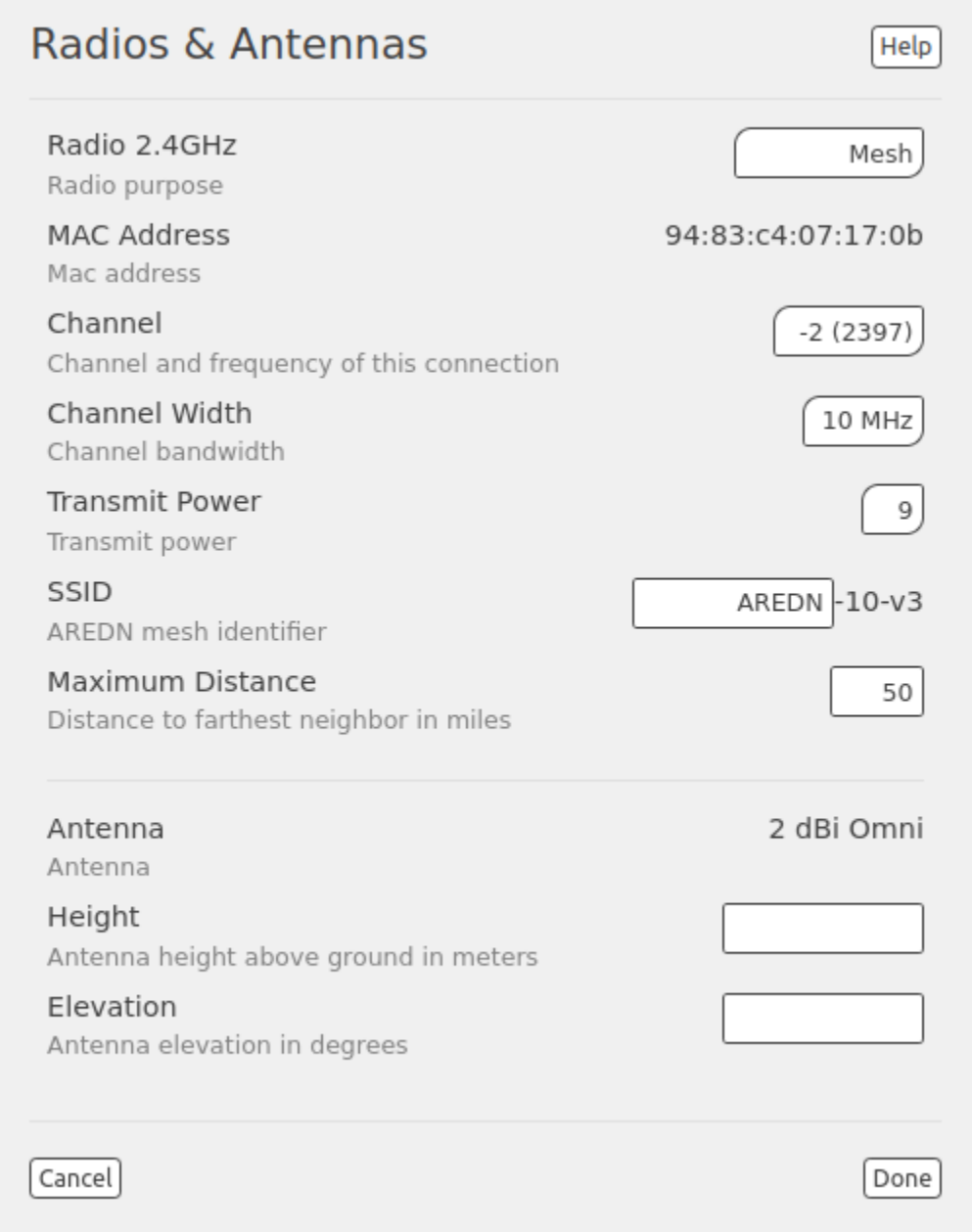
For RF nodes there is a graph of the signal level and noise floor on this link over the last hour of history (approximately). Hovering over the graph lines will display the instantaneous values which were plotted at each point on the graph. If available, a map showing the location of the node will be displayed below the graph.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` any changes.

9.13 Radios & Antennas

At the top of the right-hand column, highlight and click the section displaying your node's radio information. The **Radios & Antennas** display allows you to configure the radios on your node. Context-sensitive help is available by clicking the `Help` button.

If your device has two radios, you can configure them separately but you cannot put them both into the same mode. For example, you can use one radio for Mesh RF while the second radio functions as a LAN Hotspot or a WAN Client (as described below). Some devices may not have any available radios, but some of the radio options will still be shown if they are applicable to the device.



Click in the first field on the right to set the radio's purpose. You can choose one of several different radio functions from the dropdown list.

Mesh
: Normal AREDN® mesh mode which uses *ad hoc* peer-to-peer networking to create a mesh.

PtP/PtMP
: Uses *infrastructure* mode (point-to-point or point-to-multipoint) to limit communication between a single access point (AP) and one or more specified stations. **Mesh PtP** defines an AP that communicates with a single station. **Mesh PtMP** defines an AP that can communicate with multiple stations. **Mesh Station** defines a station that can communicate to either type of AP defined above.

LAN Hotspot
: Configures the radio as a standard 802.11 FCC Part 15 wifi AP on your node's LAN network.

WAN Client
: Configures the radio as a standard 802.11 FCC Part 15 wifi client which accesses a wifi Internet gateway for its WAN.

Off
: Disables the radio

9.13.1 Mesh settings

This option configures the radio to link with other nodes via RF across the mesh network.

MAC Address
: This displays the MAC address of the radio interface. This can be used when defining a **Mesh Station** as mentioned above and described in *Mesh Station* settings below.

Channel
: Click in the field on the right to select a channel for mesh RF communication. Nodes communicate only with other nodes that use the same channel, channel width, and SSID. You can determine the correct settings by talking with other local node operators to find out which settings are required for joining their networks. The options in this list show the channel number as well as the center frequency of each channel.

> **Warning: You are responsible for using frequencies, channels, bandwidths, and power levels that comply with your country's Amateur radio license requirements.**

Channel Width

Click in the field at the right to select from the channel widths supported on your device. Most hardware will support 5 MHz, 10 MHz, or 20 MHz channel widths, but some devices will only support specific channel widths. If the choice of channel width is limited, the device will only show its supported widths in the dropdown list.

As a general rule, a larger channel width will allow more data to be transferred, but it may only do this over shorter distances. One suggestion is to start with the largest channel width that yields a *Signal to Noise Ratio* (SNR) of at least 15 dB. There may be several reasons for reducing the channel width setting:

- To achieve a better SNR on a marginal link.
- To extend the usable distance between remote nodes.
- To increase the number of available channels in a crowded RF coverage area.

Please review the **Network Design** section for more information about designing a network that meets the specific requirements of your applications and services.

Transmit Power

Click in the field at the right to select from the power settings that are supported on your device.

SSID Setting

The default SSID is provided in the field at the right. Typically you will not need to change this default unless you have a specific reason for putting radios on a non-default SSID to filter their traffic. The SSID is analogous to a CTCSS tone; radios with different SSIDs but using the same channel may generate RF energy that causes interference, even though the radios will not be decoding each other's signals.

Maximum Distance

This is the maximum distance between remote nodes at which you can expect to achieve a usable radio link. The default value is 50 miles / 80 kilometers, but you can lower this setting if your node is only able to maintain a usable radio link with closer nodes. The distance can be limited in order to prevent distant nodes from intermittently connecting to your node due to changes in local conditions. Communicating with these distant nodes uses more radio time and can negatively impact local communications.

This distance is used by the radio when it cannot determine how far a neighbor radio is or when no radios are connected. Setting this distance appropriately is extremely important when radios are initially connecting and their location is not yet known. In particular, a value that is too low can result in radios failing to connect when they otherwise should.

9.13.2 Mesh PtMP settings

This configures the radio as a wifi *infrastructure* mode **AP** which can be accessed by one or more nodes configured as **Stations**. In this mode the SSID will include the channel being used for the links. **Station** nodes cannot communicate directly with each other but must go through the **AP**. Be aware that these links may take several minutes to initialize.

9.13.3 Mesh PtP settings

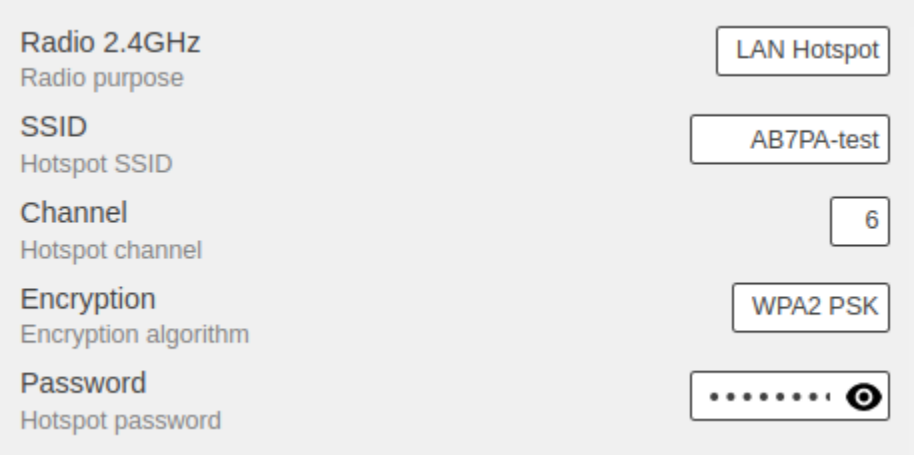
This configures the radio as a wifi *infrastructure* mode **AP** which can be accessed by a single designated **Station** node. In this mode the SSID will include the channel being used for the link, and a field appears which specifies the MAC Address of the **Station** node that is allowed to peer with this PtP **AP**. Be aware that these links may take several minutes to initialize.

9.13.4 Mesh Station settings

This configures the radio as a wifi *infrastructure* mode **Station** which can connect to a designated PtMP or PtP **AP** node. In this mode the SSID will include the channel being used for the link. Be aware that these links may take several minutes to initialize.

9.13.5 LAN Hotspot settings

This configures the radio as a standard 802.11 FCC Part 15 wifi hotspot on your node's LAN network. Any device that connects to your node using this wifi AP will receive an IP address on your node's LAN subnet.



SSID
: A default SSID is provided, but you should change this value to a unique name that identifies the hotspot to potential users who will connect to it locally.

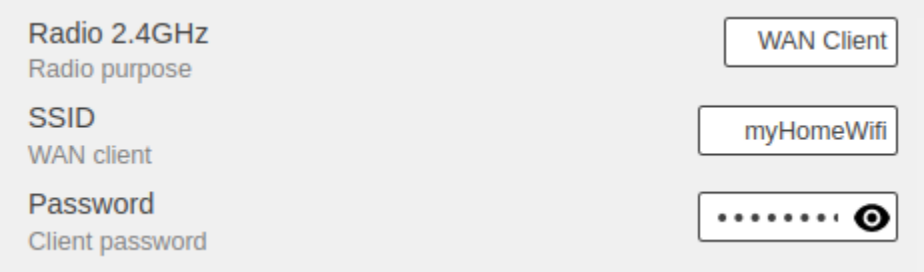
Channel
Click in the field to the right to select a valid wifi channel. You are responsible for using a channel that complies with your region's wifi requirements (for example, FCC Part 15).

Encryption
Click in the field to the right to select a wifi encryption method.

Password
Click in the field to the right to enter a valid wifi password for accessing your node's hotspot. You can click the *eye* icon at the right of the password fields to toggle between hidden and visible text.

9.13.6 WAN Client settings

This configures the radio as a standard 802.11 FCC Part 15 wifi client which can connect to a wifi AP. This is used to provide WAN Internet access to your node via wifi rather than requiring an Ethernet cable plugged into the node's WAN port. Enabling a radio as a *WAN Client* will disable VLAN1 on your node, so Internet access will no longer be possible through the physical WAN port.

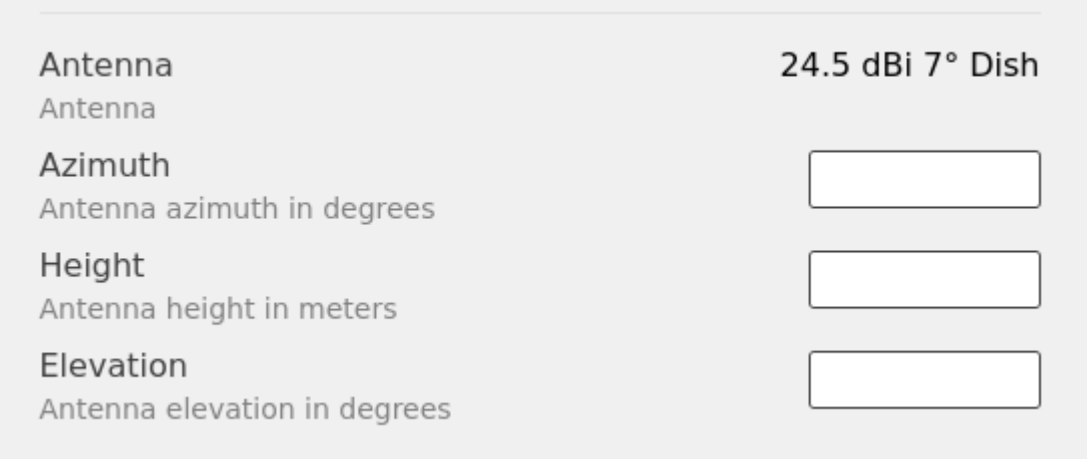


SSID
Click in the field at the right to enter the SSID of the local wifi access point you are connecting to for Internet access. Set your node's WAN interface to receive an IP address via DHCP from the wifi AP which will provide Internet connectivity.

Password
Enter the authentication password for the wifi AP to which you are connecting. Your node uses *WPA2 PSK* encryption to connect to external wifi APs. The password length must be between zero and 64 characters. If the key length is 64, it is treated as hex encoded. If the length is 0, then no encryption will be used to connect to an open AP. A single-quote character (') must not be used in the passphrase. You can click the *eye* icon at the right of the password fields to toggle between hidden and visible text.

9.13.7 Antenna settings

Various devices may have differing antenna configurations, so the appropriate fields will be displayed depending on your radio hardware. If there are multiple antenna types available for your hardware model, then you can select one from a dropdown list.



Azimuth
: Click in the field at the right to enter the direction (in degrees) toward which your directional antenna is aimed. This field will not appear if your device uses an omnidirectional antenna.

Height
: Click in the field at the right to enter a height in meters above ground level at which you have your antenna mounted.

Elevation
: Click in the field at the right to enter an angle (in degrees) of uptilt or downtilt that you have set on your antenna. Note that some omnidirectional and sector antennas have a built-in downtilt, and that value can be entered here.

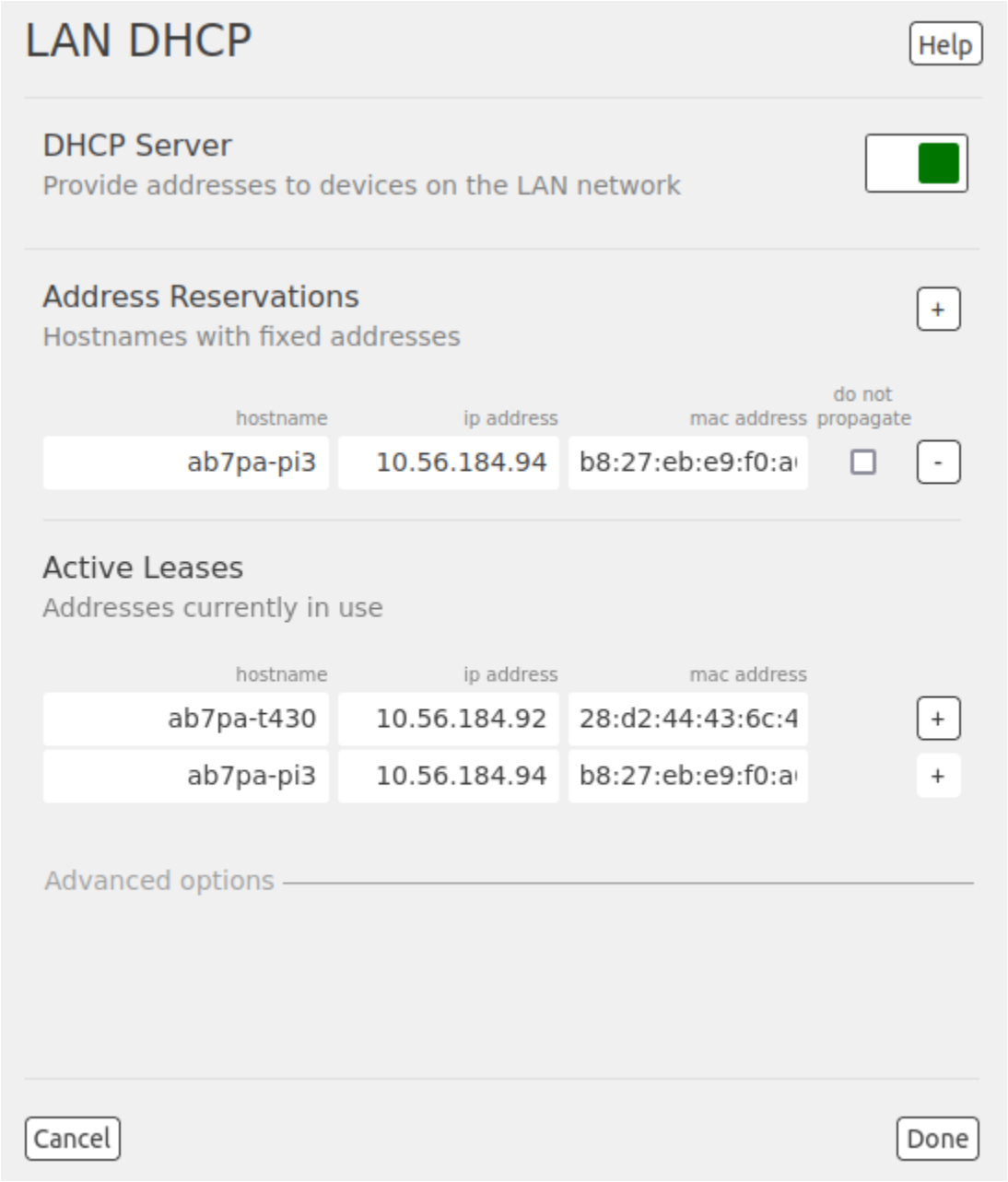
You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` any changes.

9.14 Mesh section

This section displays summary statistics that include the number of nodes, devices, and services currently visible from this node. When you hover over the *Mesh* section, a gray background appears which indicates that this section is selected. If you click in the section, you will be taken directly to the **Mesh Status** display.

9.15 LAN DHCP settings

Highlight and click the section displaying your node's *LAN DHCP* settings. By default each node runs a Dynamic Host Control Protocol (DHCP) server to provide client IP addresses for devices joining its LAN network. LAN devices connecting to your node will be assigned an IP address automatically. Context-sensitive help is available by clicking the `Help` button.



9.15.1 DHCP Server

This option is `enabled` by default, which provides IP addresses to devices attached to this node's LAN network. If disabled, the LAN network is still active, but addresses will not be automatically provided. Multiple DHCP servers can be active on the same LAN network but it is not defined which DHCP server will provide an IP address to each device even when address reservations are configured. It is best practice to have only one DHCP server enabled on a LAN network in order to avoid confusion.

9.15.2 Address Reservations

Devices which are added to the *Address Reservations* list will display their hostname, IP address, and MAC address. The hostname of every device connected to the mesh at large should be unique. It is best practice to prefix your Amateur Radio callsign to the hostname of each of your devices in order to give it a unique name on the network.

You can create an *Address Reservation* by clicking the [+] icon to the right of the **Address Reservation** title. Click in the first field to enter the new device's hostname. In the second field select an unused IP address from the dropdown list. In the third field type the MAC address of the new device. If you have a device which needs to be reachable via your node, but which should not be accessed across the mesh network, click the *Do Not Propagate* checkbox to prevent the routing protocol from propagating that information across the mesh.

There may be some devices on which you are not able to set the hostname, but once you add that device to your *Address Reservations* you can click in the *hostname* field to edit the hostname that will be propagated across the mesh. You may also want to assign a specific IP Address to the device by selecting it from the drop-down list. You can click the *Do Not Propagate* checkbox to prevent the routing protocol from propagating the new device's information across the mesh.

In addition to adding an address reservation manually, you can also click the [+] icon at the right of any of the devices which have active DHCP leases as described below. You will then see that host appear in the *Address Reservations* list.

9.15.3 Active Leases

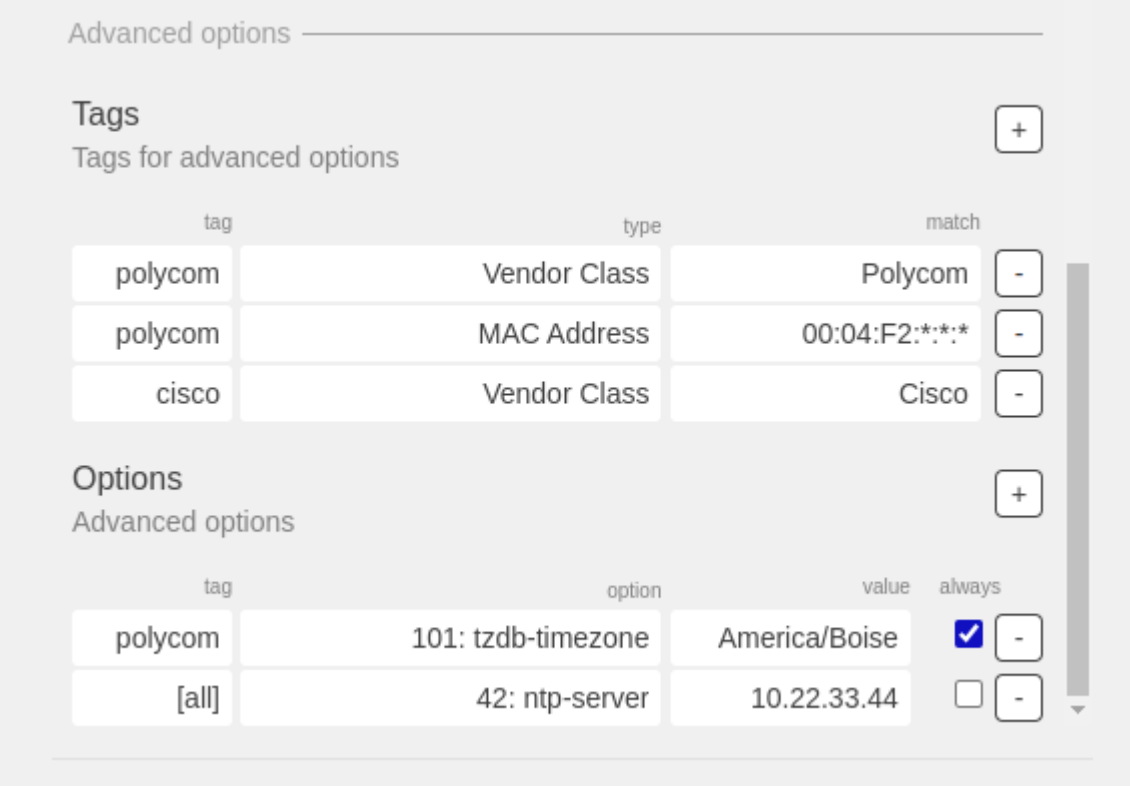
Devices which are currently assigned an IP address by your node will be displayed in the table of *Active Leases*. The first field displays the hostname, followed by the IP address that was assigned by your node's DHCP server. The third field displays the device's MAC address.

Since DHCP leases are dynamic and can change over time, there may be a reason why a host's assigned IP address should be made permanent. This is especially useful if that host will provide an application, program, or service through your node to the mesh network at large. As mentioned above, you can reserve that host's DHCP address by clicking the [+] icon at the right of the row. You will see that host now appear in the *Address Reservations* list.

9.15.4 Advanced DHCP Options

Additional options will be displayed when you click **Advanced Options**. This section allows you to specify DHCP option codes and values which are sent to devices on your node's LAN network. In addition to providing an IP address, the DHCP protocol is able to send a large number of options for device configuration. Any LAN client joining the network can request specific DHCP options in addition to its IP address. These *Advanced Options* are especially helpful for configuring and provisioning VoIP phones on your node's LAN.

The Internet Assigned Numbers Authority (IANA) is a good source of information about DHCP options. Specific vendor equipment may or may not support all of the options, so you should verify which options are supported by referring to the manufacturer's documentation for your LAN device.



Tags

The tags for advanced DHCP options allow you to define labels for values that will be assigned to clients which match specific properties such as Vendor Class or MAC address. Click the [+] icon to add a new tag. Enter a tag label in the first field, then click in the second field to select a tag type from the dropdown list. Finally, enter a text string which will be used to match a property on the device, such as the Vendor Class or MAC address. To delete an existing tag, click the [-] at the right side of the row you wish to remove.

Options

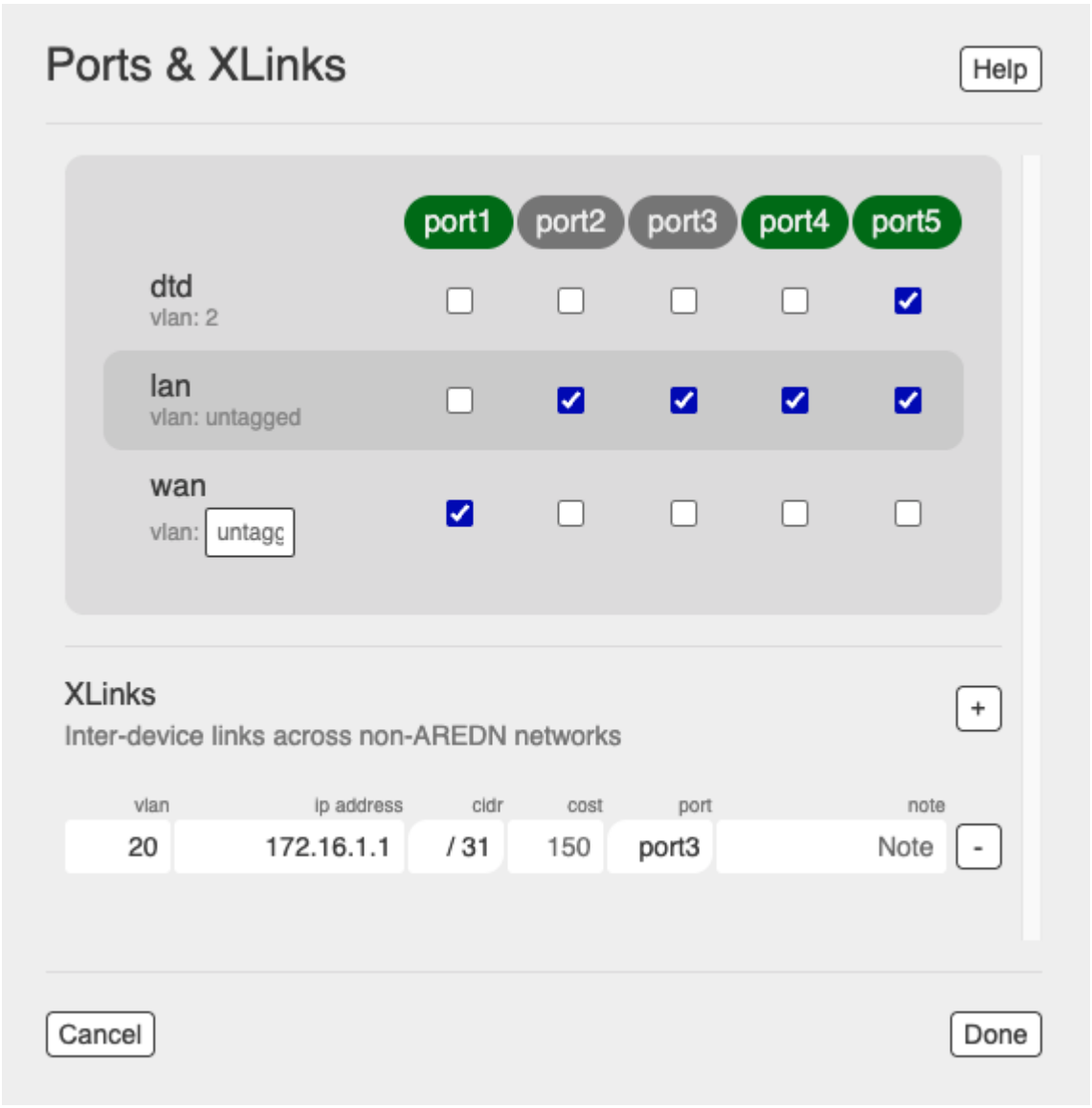
The options entries allow you to specify which devices will receive the DHCP options. Click in the first field to select whether you want this option to be sent to [all] clients or only to clients which match a specific tag. Option numbers can be entered directly in the second

field or you can select them from the dropdowm list of well-known options. In the third field enter the specific value that will be sent in this option. A checkbox allows you to specify whether or not this option will always be sent.

To delete a tag or option, click the [-] icon on the right of the existing row for the item you wish to delete. You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` any changes.

9.16 Ethernet Ports & Xlinks

If you have a multiport node or one which supports xlinks, then the *Ethernet Ports & Xlinks* section will be displayed. This provides a way for you to configure the ports on your node and/or the configuration of xlinks. Context-sensitive help is available by clicking the `Help` button.



Ports (if available)

The *Ports* section shows a table of the available port names at the top of each column, with configuration labels for each row along the left side, and checkboxes beneath the ports to show which settings have been assigned on each port. For more information about the standard AREDN® VLANs, refer to the *VLAN* description in the *Advanced Options* section of **Network** settings.

The example configuration shown above is for a *Mikrotik hAP ac2/ac3*.

- The first port is configured with the WAN checkbox selected. The data entry field to the right of the *vlan* label can contain any valid vlan identifier if it is required. The default for the multiport node in this example is no vlan (untagged). Leave the default value unless there is a specific reason why it must be changed for your situation.
- The remaining ports in this example are identified as LAN ports. The middle ports have no special settings (untagged), but the last LAN port is configured as a DtD link port

which will have an Ethernet cable connecting it to another AREDN® node.

If you want to change a port's configuration, simply check or uncheck the settings desired on each port.

Xlinks

A cross-link (xlink) allows your node to pass AREDN® traffic across non-AREDN® links. To add an xlink click the [+] icon, enter an unused VLAN number for the link. Enter the IP address for the link, the CIDR netmask, and a weighting factor which will be used by the routing protocol to determine the best route for AREDN® traffic. On a multiport device you also enter the port to which the near-side device is connected to your node. You may also add a note to describe the link or provide contact information. If you want to remove an xlink, simply click the [-] icon on the right side of the row to remove it.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` any changes.

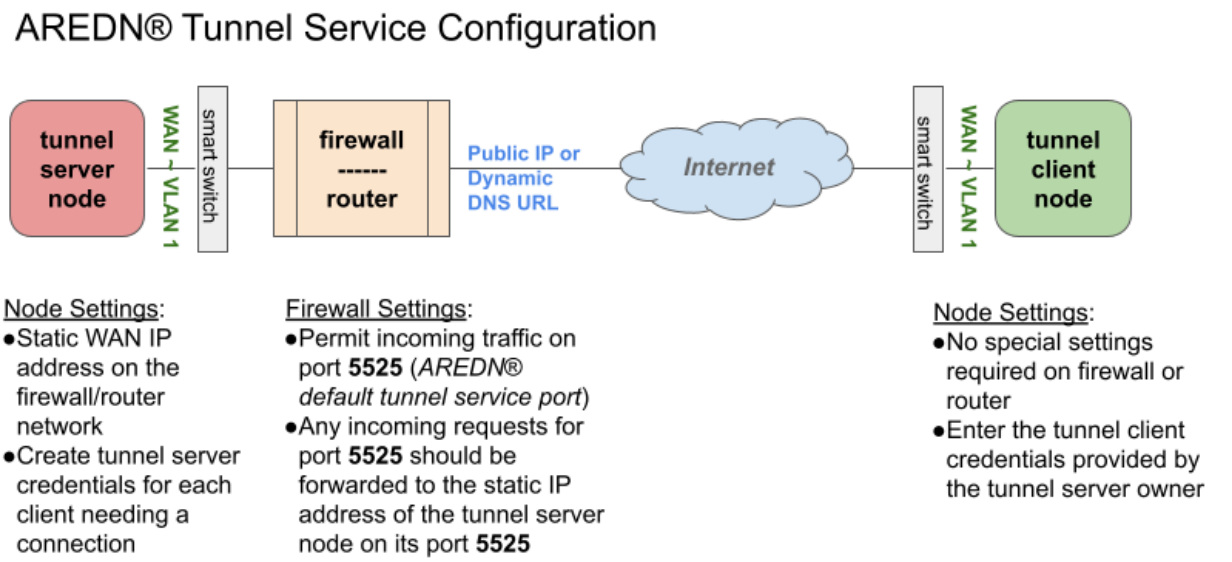
9.17 Tunnels

Tunnels are typically used as a means of connecting mesh islands if RF links cannot be established. Before using the AREDN® tunnel feature, be aware of how this type of connection could impact your local mesh network. If your node participates in a local mesh, then adding one or more tunnel connections will cause the nodes and hosts on the far side of the tunnel(s) to appear as part of your local mesh network. This essentially joins the two networks into a single large network, increasing the total network traffic across the entire range of devices.

If you want to participate in remote mesh networks, consider using the *Cloud Mesh* network established through worldwide Supernodes. If your local network does not have a Supernode and you need to connect to another remote network, consider establishing a tunnel from a standalone node that is *not* connected to your local mesh. Remember that AREDN® is first and foremost an emergency communication resource, so it's possible that Internet-dependent links and the assets they provide will not be available during a disaster or deployment.

9.17.1 Internet Networking Requirements

In order to run your node as either a *Tunnel Server* or *Tunnel Client*, you will need to configure Internet access. The following diagram shows an example of tunnel connectivity between two nodes using network port `5525` as an example.



Multiport nodes have the appropriate VLANs preconfigured in the AREDN® firmware. If you are using any other type of node, then you will need to configure a separate VLAN-capable switch. Set your VLAN-capable network switch to appropriately tag traffic from the Internet with *VLAN 1* before sending it to your node. This allows your node to properly identify the traffic as coming from the Internet to its WAN interface. See the equipment manual for your smart switch to determine how to configure VLAN settings.

Tunnels allows you to configure connections for tunnel roles (Client & Server). The Wireguard tunneling protocol provides an *encrypted* UDP (User Datagram Protocol) connection that is both efficient and secure. It only encrypts the traffic as it traverses the public Internet, so no encrypted traffic will be sent via radio in compliance with FCC Part 97 requirements.

Attention: Any older legacy *vtun* tunnels should be migrated to Wireguard as soon as possible.

Networking for Tunnel Servers

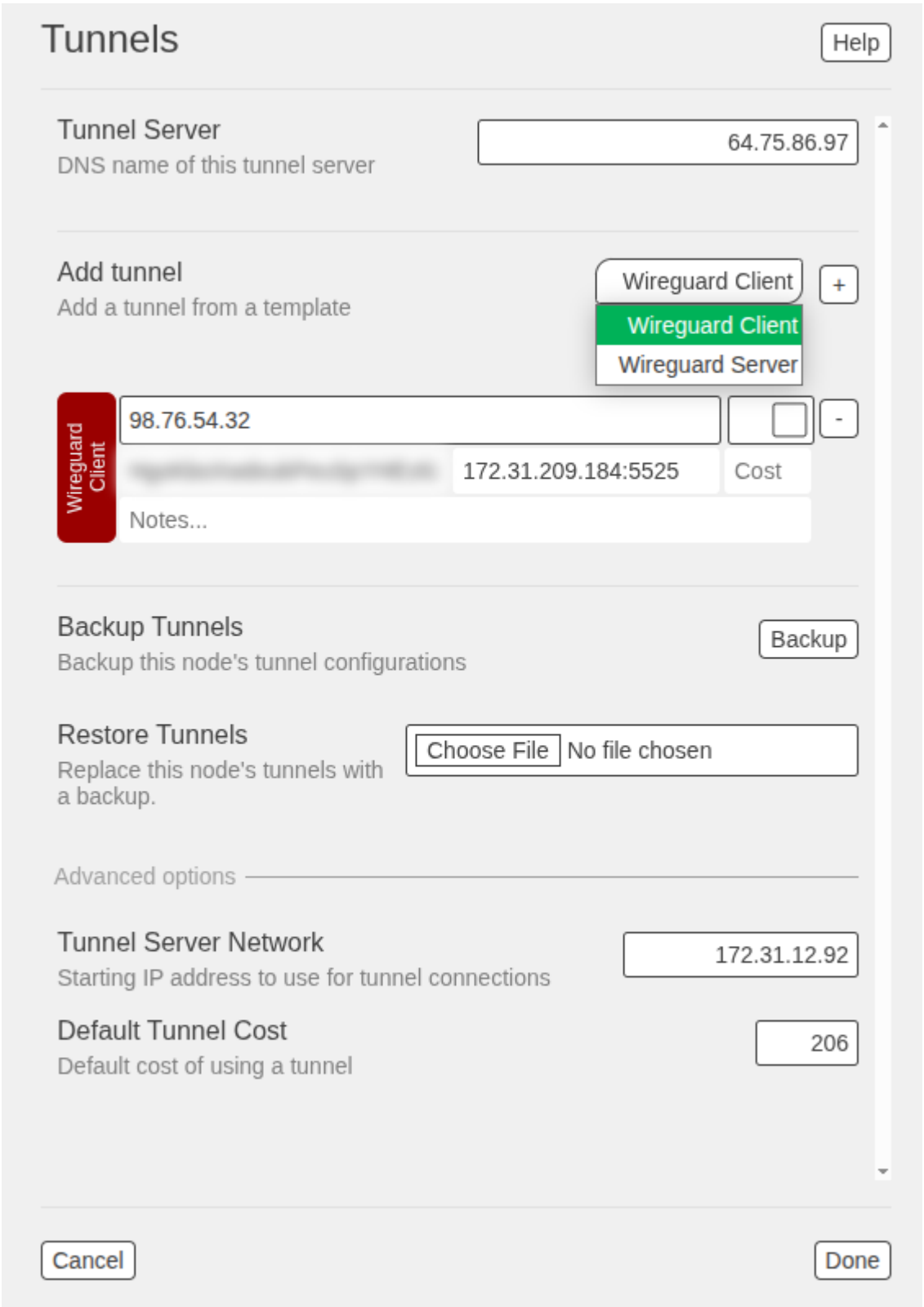
In order for remote tunnel clients to reach your tunnel server node, your Internet-connected firewall must allow that traffic to enter your network and it must also forward that traffic to your tunnel server node. In order for your router/firewall to have a consistent way to forward traffic to your node, it is best practice to set a static IP address on your tunnel server node's WAN interface or to reserve its DHCP IP address in your router.

On your Internet-connected router/firewall set the firewall rules to permit UDP traffic from the Internet on an appropriate range of ports. The starting port should be `5525`, which will provide for one Wireguard tunnel client connection. If you want to allow up to 10 Wire-

guard tunnel links (for example), you would permit UDP traffic on the range of ports between `5525-5534`. Then configure a port forwarding rule to send any traffic from the Internet on your range of ports to the IP address of your node's WAN interface.

9.17.2 Tunnel settings

Highlight and click the section displaying your node's **Tunnels** to open the tunnel configuration display as shown below. Context-sensitive help is available by clicking the `Help` button.



Tunnel Server

This first setting is relevant if you will be using your node as a tunnel server. Otherwise

you can skip to the next section. A tunnel server node must be reachable from the Internet. Enter the public IP address (obtained from your ISP (Internet Service Provider)) or DDNS hostname in the field at the right.

9.17.3 Add Tunnel

To add a tunnel connection, click in the field at the right to select from the dropdown list the type of tunnel you want to create. Be aware that without proper time synchronization, Wireguard will not establish tunnels. Make sure that an NTP or GPS time source is reachable at boot time so that the key exchange between the client and server will happen correctly. If mesh based NTP servers are available, ask the owners to advertise them as services to ensure that time synchronization happens across your mesh network even if the Internet is not available. Review the **Local Services** section above for instructions on advertising a local NTP server.

For each tunnel definition there is a *Cost* or tunnel weight field. The global default tunnel cost is configured under *Advanced Options* as described below, but you can override this value on a per tunnel basis. Leave this field empty to accept the global default, or enter a tunnel cost to override the default if you desire. Each tunnel definition also has a *Notes* field in which you may enter helpful notes about the tunnel link.

Wireguard Client

Select *Wireguard Client* from the dropdown list and click the [+] icon. For tunnel client credentials, contact the Amateur Radio operator who controls the tunnel server you want to connect to and request client credentials by providing your specific node name. The tunnel server administrator will send you the public IP or hostname for the tunnel server field, the key you are to use, and the network IP address & port for your client node. If your client credentials were provided using the method described below for servers, you can highlight and copy the entire set of values, click into one of the fields on your tunnel client row, and when you paste into one of the fields then all of the credentials will be automatically entered into the correct fields for you. Otherwise, you can manually enter these values into the appropriate fields on your node.

Wireguard Server

Select *Wireguard Server* from the dropdown list and click the [+] icon. In the `Node Name` field enter the exact node name of the client node that will be allowed to connect to your tunnel server. Do not include the "local.mesh" suffix. The security key, network, and port settings are automatically generated and displayed. Click the *copy* icon to the right of the *Notes* field to display all of the connection settings in a new web page. These settings can then be copied and pasted into an email or text file to provide the credentials to the owner of the client node.

The switch on the right is `enabled` by default, but it appears gray until the tunnel connection is established, at which time it will be green.

Tunnel Backup/Restore

If you want to keep a copy of all your tunnel settings you can click the `Backup` button to save them to a file. This will backup the tunnel credentials and settings as well as the DNS value and tunnel subnets. If you have a previously saved tunnel backup file, you can restore those settings to your node. Choose the tunnel backup file to restore and those setting will be displayed. Click `Done` and you will be returned to the *admin status* page where you can `Commit` or `Revert` the changes. The settings in the tunnel backup file will overwrite and replace any existing settings on your node.

Advanced Tunnel Options

The **Tunnel Server Network** address is displayed under *Advanced Options*. This is the starting IP address for your tunnel server's network, and it is calculated automatically. It should not be changed unless there is a specific reason why the default will not work for your situation.

> **Attention:** This value is only editable when there are no existing server credentials being provided by your node. If you already have tunnels on your node then this field will be grayed out and uneditable.

The **Default Tunnel Cost** is the weighting factor used by the routing protocol to determine the link cost of sending traffic via the tunnel. This value is a global default, but you can override the tunnel cost by providing an individual per-tunnel value as described above.

You can click the `Cancel` button to ignore any changes you made on this display. When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` any changes.

9.18 Tools

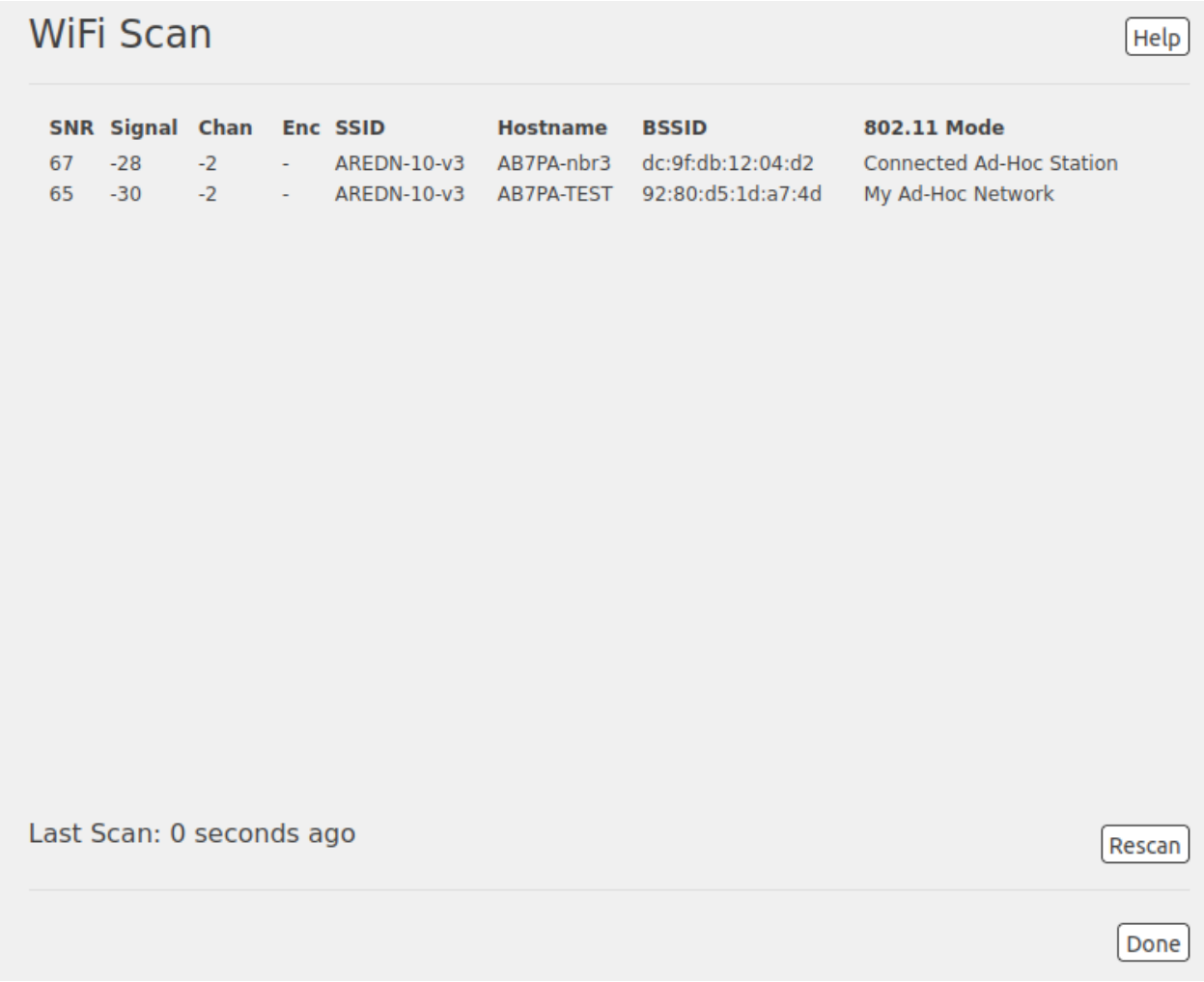
 Click the **Tools** icon at the bottom of the left nav bar and select one of the tools from the popup menu.

For any tools with dropdown selection lists, you may filter the list by typing characters in the search box. This will limit the list to include only items which match the text you enter. As you type each character from your keyboard into the search field, the list will change to show only the entries that match your character string. The filter is case insensitive, so it will find both uppercase and lowercase entries for the characters you enter.

9.18.1 WiFi Scan

This displays the *wifi scan* page which will show the results of the most recent scan (if any). It will only appear if the radio is in Mesh mode. Context-sensitive help is available by clicking the `Help` button.

Click the `Scan` button in the lower right corner to initiate a new scan which looks for wifi signals that are using the same channel width as your node. It is best practice to scan on 5, 10, and 20 MHz channel widths to find any 802.11 signals within range. Several scans may be necessary to find as many local devices as possible.



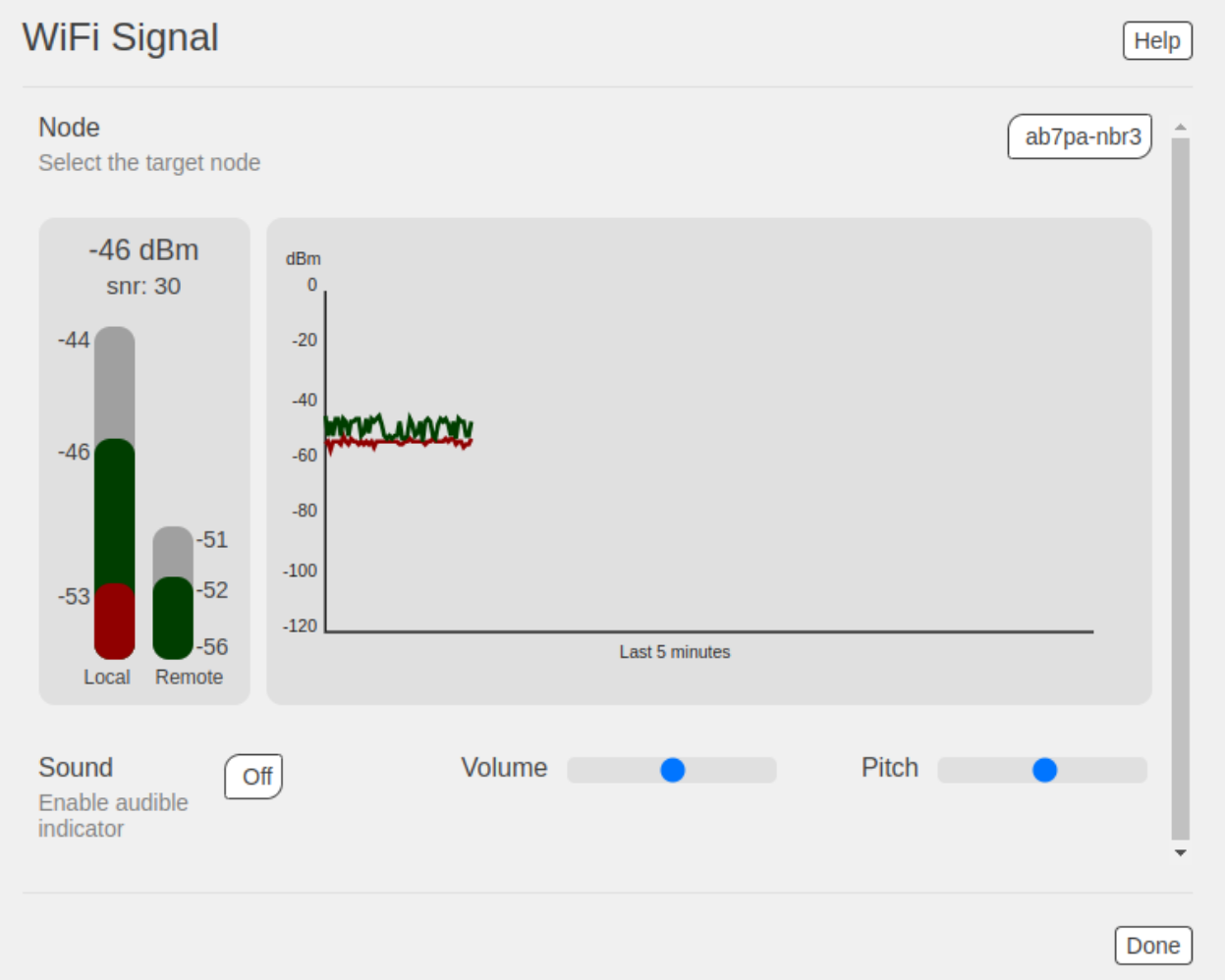
Note: The BSSID column shows the IEEE 802.11 wireless *Basic Service Set ID*. This is a 48-bit label that conforms to the MAC-48 convention, but it is not an actual MAC address. More information about the BSSID can be found here.

With some devices, a scan will momentarily disconnect the wifi from the mesh so the radio is available to perform the scan operation. It is recommended that you perform a scan when connected

to the device in some other way than via WiFi. The most recent scan results are retained. When you are finished studying the scan results, click the `Done` button to return to the *admin* display.

9.18.2 WiFi Signal

This displays RF signal information as a realtime line graph. It will only appear if the radio is in Mesh mode. The default view shows the average signal of all connected stations in realtime. Click in the field to the right of the *Node* label to select a specific neighborhood node from the dropdown list. The graph will be cleared and redrawn using signal data from that node. Context-sensitive help is available by clicking the `Help` button.



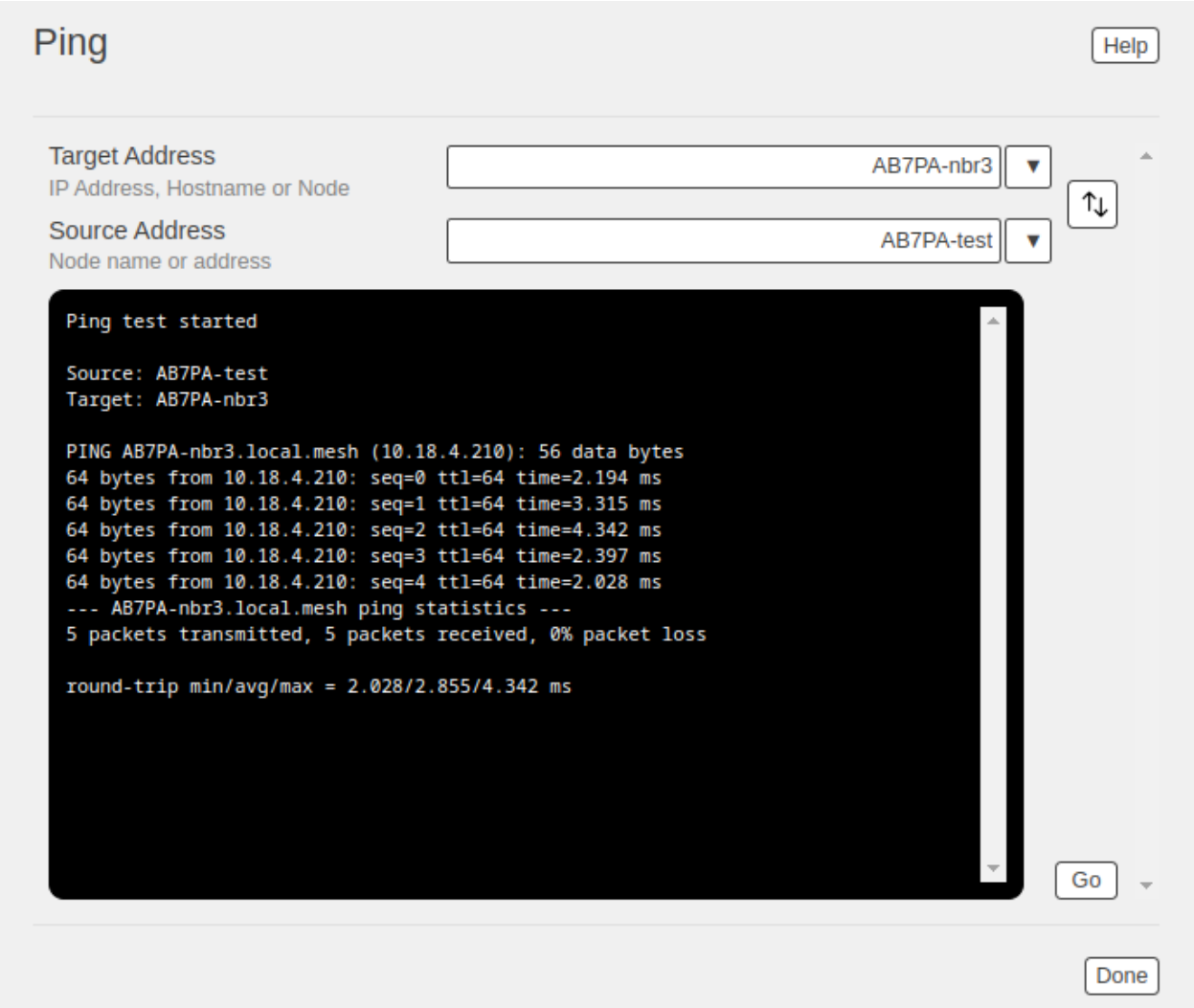
The colored bars on the left display the worst and best signal values that are seen during the monitoring interval. The instantaneous signal value is shown above the colored bars on the left. Both the

local node and remote node view of the signal levels will be displayed on the bars and the graph. All of these values will be adjusted over time as new data is obtained.

Below the line graph there are controls that allow you to enable an audio representation of the instantaneous signal value. Click in the field to the right of the *Sound* label and select `OFF` or `ON` to enable or disable the sound. You can control the volume and pitch of the tone using the horizontal sliders. The higher the pitch, the better the signal level. When you are finished studying the results, click the `Done` button to return to the status display.

9.18.3 Ping

This tool allows you to perform a ping test between devices on your network. Context-sensitive help is available by clicking the `Help` button.



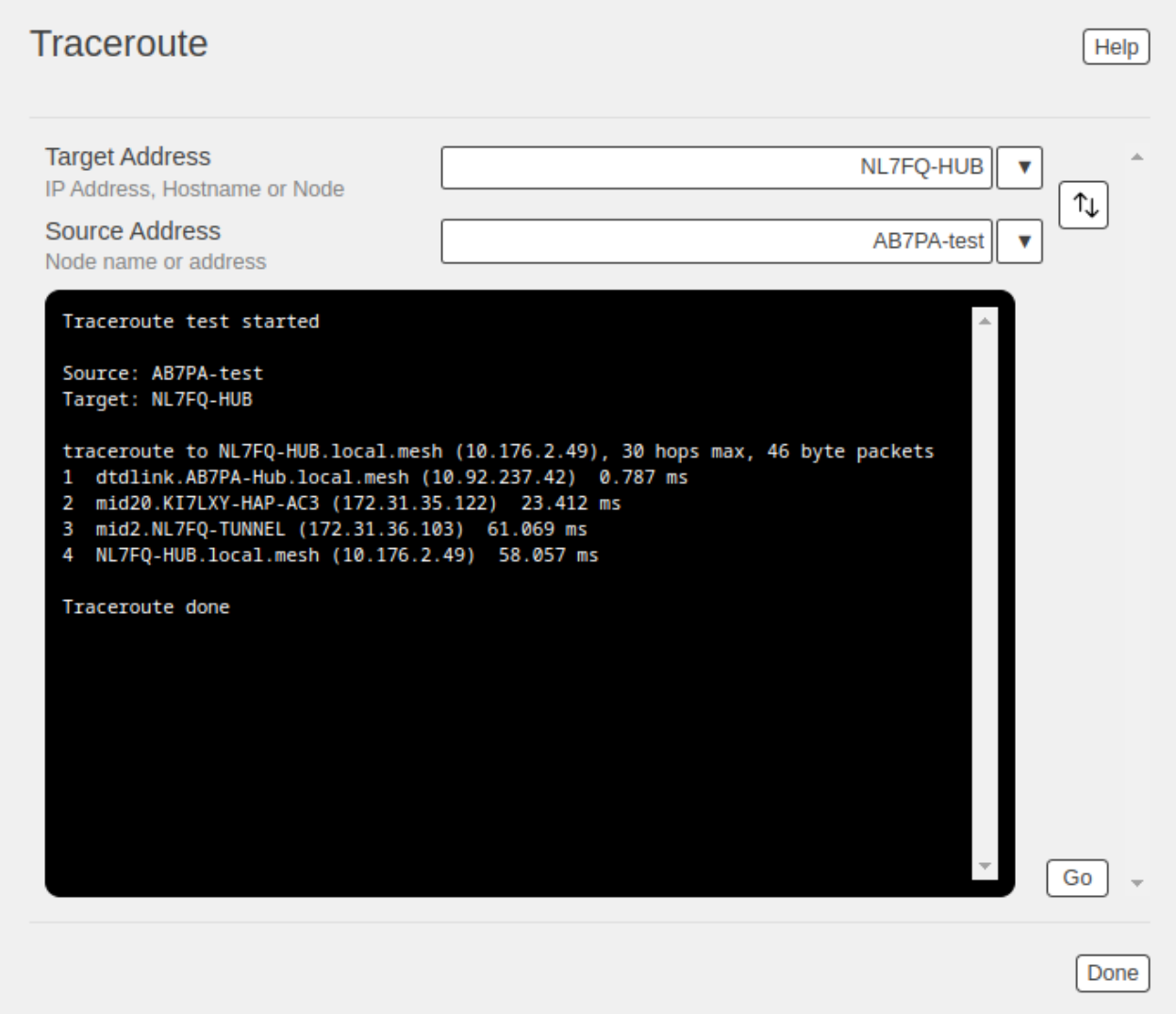
Target Address
: Click the down arrow icon at the right of the *Target Address* to select a target device from the dropdown list. If your desired device is not shown, you can click in the field to enter or edit the hostname or IP address that you want to use as the target. This can be any device or address which is capable of responding to pings.

Source Address
: The *source* must always be an AREDN® node, and by default the current node name is automatically entered. Click the down arrow icon at the right of *Source Address* to select a node from the dropdown list. If your desired node is not shown, you can click in the field to enter or edit the node name that you want to use as the source.

After selecting the *Target* and *Source*, click the `Go` button in the bottom right corner to view the results. You may want to test network connectivity in both directions by clicking the double-arrow icon to swap the *Target* and *Source* devices, remembering that your *source* must always be an AREDN® node. When you are finished studying the results, click the `Done` button to return to the status display.

9.18.4 Traceroute

This tool allows you to perform a traceroute between two devices on your network. Context-sensitive help is available by clicking the `Help` button.



Target Address

Click the down arrow icon at the right of the *Target Address* to select a target device from the dropdown list. If your desired device is not shown, you can click in the field to enter or edit the hostname or IP address that you want to use as the target.

Source Address

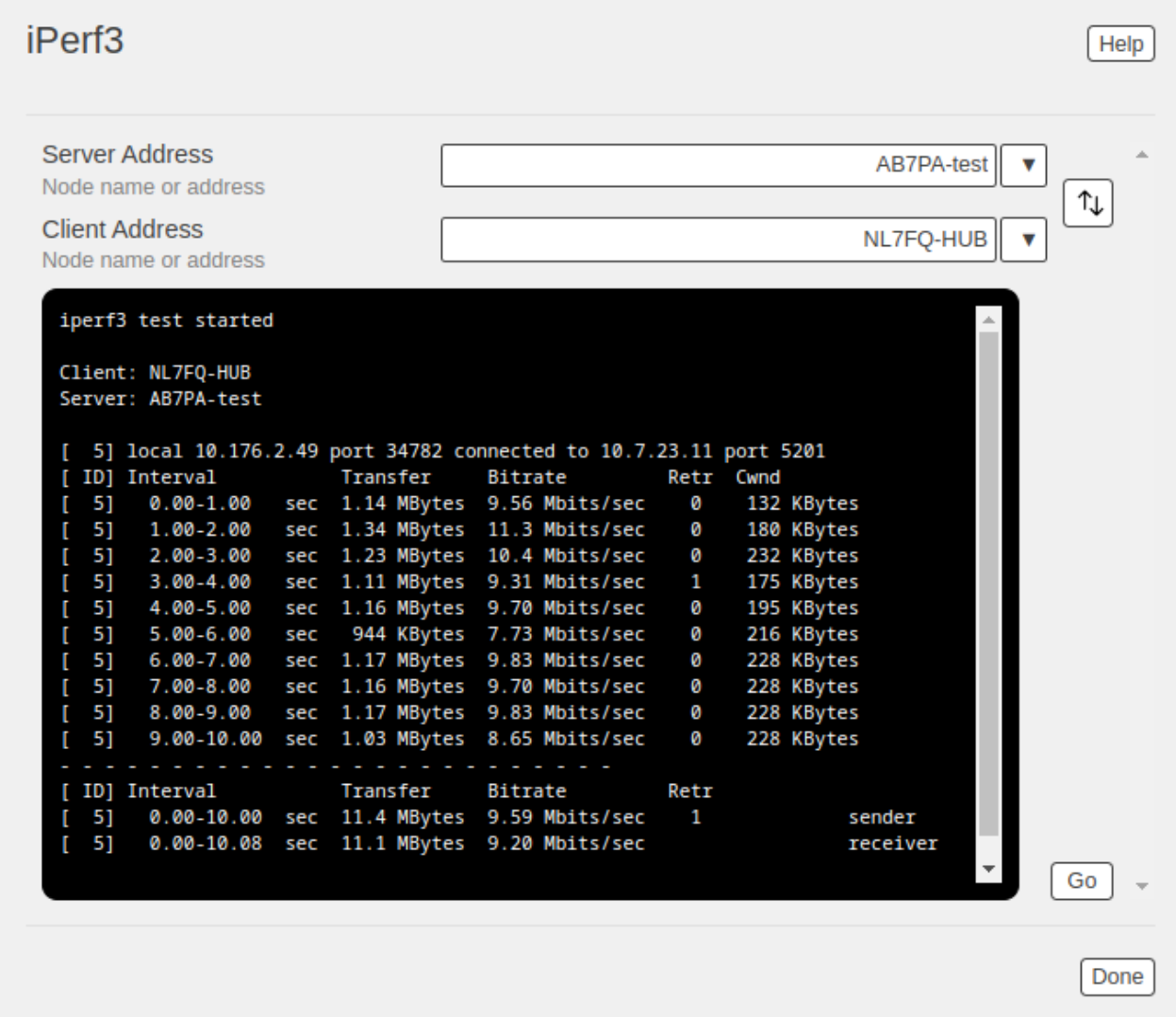
The *source* must always be an AREDN® node, and by default the current node name is automatically entered. Click the down arrow icon at the right of *Source Address* to select a node from the dropdown list. If your desired node is not shown, you can click in the field to enter or edit the node name that you want to use as the source.

After selecting the *Target* and *Source*, click the `Go` button in the bottom right corner to view the results. You may want to test network connectivity in both directions by clicking the double-arrow

icon to swap the *Target* and *Source* devices, remembering that your *source* must always be an AREDN® node. When you are finished studying the results, click the `Done` button to return to the status display.

9.18.5 iPerf3

This tool allows you to perform throughput tests between two AREDN® nodes on your network using iperf3. Context-sensitive help is available by clicking the `Help` button.



Server Address

Click the down arrow icon at the right of *Server Address* to select a node from the dropdown

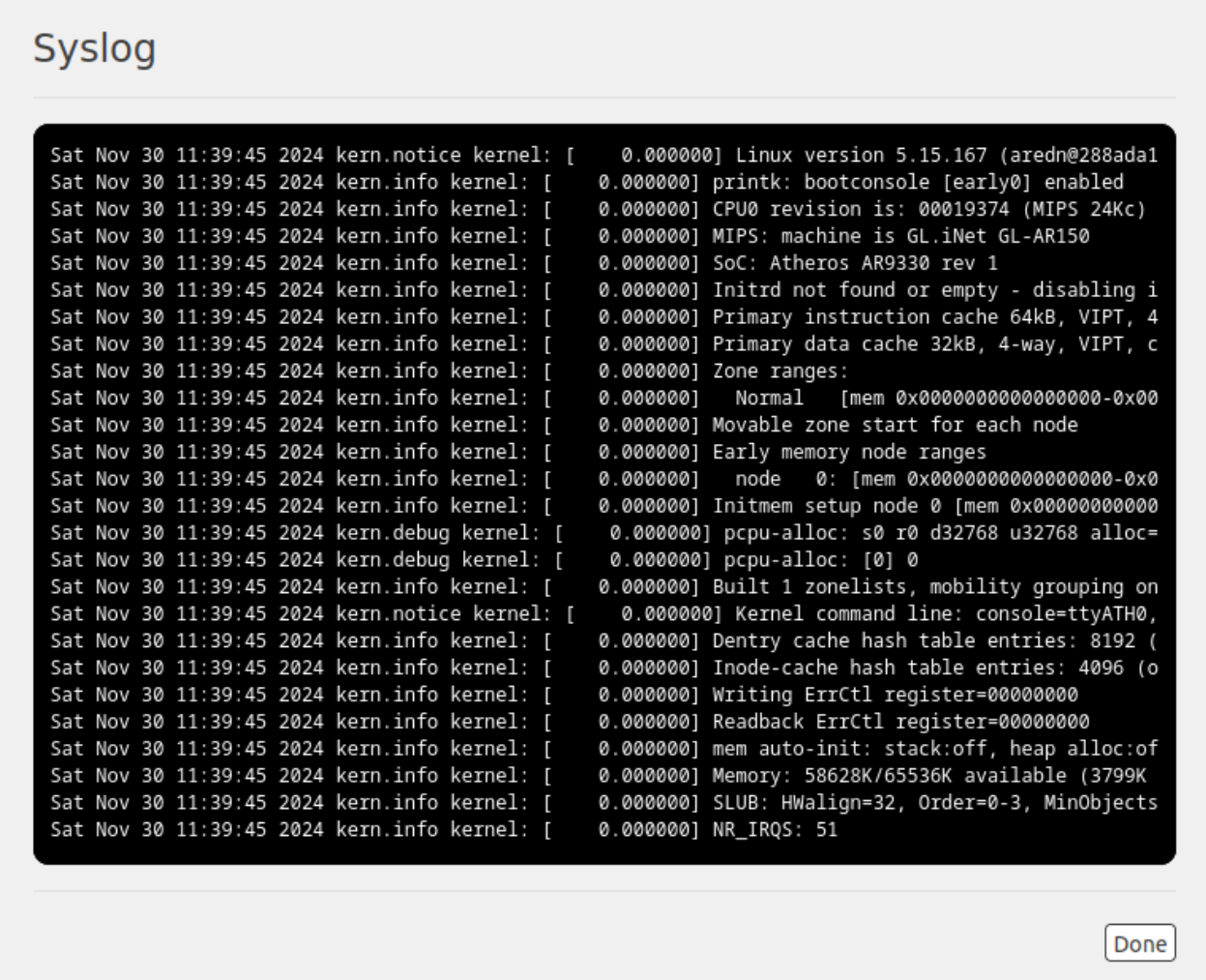
list. If your desired node is not shown, you can click in the field to enter or edit the node name that you want to use as the iperf3 server.

Client Address
: By default the current node name is automatically entered as the client, but you can click the down arrow icon at the right to select any node from the dropdown list. If your desired node is not shown, you can click in the field to enter or edit the node name that you want to use as the client.

After selecting the *Server* and *Client* nodes, click the Go button at the lower right corner to view the results. You may want to test network throughput in both directions by clicking the double-arrow icon to swap the *Server* and *Client* nodes. When you are finished studying the results, click the Done button to return to the status display.

9.18.6 Syslog

This tool allows you to view the log file on your node. You can scroll up and down through the log entries as needed. When you are finished studying the results, click the Done button.



9.18.7 Support Data

There may be times when you want to view more detailed information about the configuration and operation of your node, or even forward this information to the AREDN® team in order to get help with a problem. Click the *Support Data* icon to save a compressed archive file to your local computer. It is also possible to generate a support data file from the command line of your node:

```
# /usr/local/bin/supportdata
Generated support data file: /tmp/supportdata.tar.gz
```

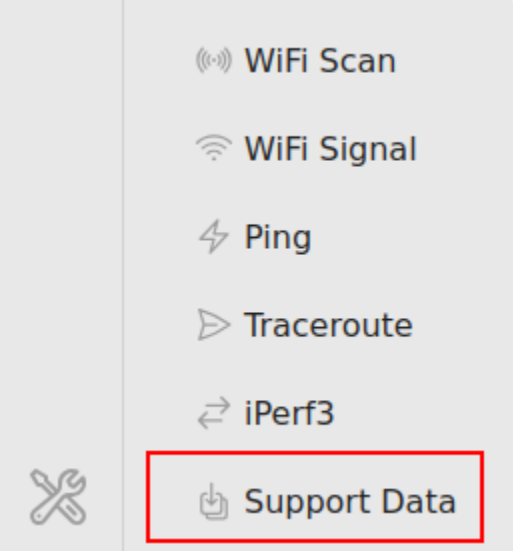
CHAPTER

TEN

REPORTING PROBLEMS OR ISSUES

If you experience issues with building or using AREDN® devices, there are several sources of help. There is an active user community that regularly contributes to the AREDN® Forum, and you can post your experience there to receive help and feedback.

However, if you have issues that you think should be investigated by the AREDN® development team, you can follow the steps below for engaging with the software developers.



Download a Support Data File

Every node has a built-in tool that allows you to download a support data file containing information that is helpful for troubleshooting. To download a support data file from your node, login as your node *admin* and click the tools icon in the left nav bar. In the list of available tools, click the **Support Data** link and the support file will be downloaded to your computer.

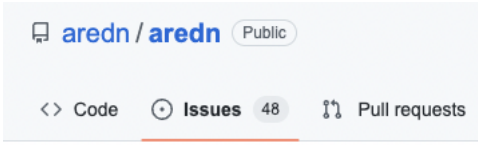
Create a GitHub account

To open an issue on GitHub you first must create your own GitHub account. This is free and easy to do by following these steps:

1. Open your web browser and navigate to the GitHub URL.

2. Click the `Sign Up` button and enter the required information. We suggest using your callsign as the username.
3. On the GitHub website, click the `Sign In` button and authenticate to GitHub with the credentials you created.
4. Navigate on GitHub to the AREDN® code repository: `https://github.com/aredn/aredn`

Open a new issue on GitHub



There are several sections in the *aredn/aredn* code repository, and you can navigate to the issues area by clicking **Issues** in the top horizontal menu.

1. To open a new issue click the **New Issue** button on the upper right side.
2. Enter a meaningful title in the *Title* field.
3. Use the edit box to describe your issue fully. You should include the exact hardware model and firmware version on which you saw the issue.
4. You can attach screenshots or support data files by dragging and dropping them into the text window.
5. Click the **Submit New Issue** button to submit the issue for review.

Once the issue is submitted you can click the title in the issues list to see the details. You can enter additional information as a new comment on the existing issue. When any future comments or questions are posted to your issue you will receive notifications of those updates. If the issue has been resolved, you can then close your issue if you desire.

CHAPTER

ELEVEN

NETWORKING OVERVIEW

This **Network Design Guide** will discuss some of the useful principles for creating robust radio networks as a service both to the amateur radio hobby and the community at large. An AREDN® network is able to serve as the transport mechanism for the applications people rely upon to communicate with each other in the normal course of their business and social interactions, including email, chat, phone service, document sharing, video conferencing, and many other useful programs. Depending on the characteristics of the implementation, this digital data network can operate at near-Internet speeds with many miles or kilometers between network nodes.

There are a variety of ways to interconnect AREDN® nodes, but the most important question that should be answered is *"What is the purpose for this particular network?"* The specific requirements of your situation will drive the design of your data network. For example, consider the following issues.

Temporary or Permanent

Is your network being deployed as a short-term communication mechanism, possibly to meet the needs of a day-long event or a training exercise? If so, then several amateur radio operators with portable nodes can quickly establish an *ad hoc* mesh network with a specific set of services to meet the communication needs for that situation. Those nodes and computers can probably operate from portable batteries, without any external power dependencies for such a limited-time deployment.

Is your network intended as a long-term or permanent infrastructure to serve the on-going communication needs of a local region? If so, then a more sophisticated network topology must be designed and constructed to meet those long-term requirements. More robust or ruggedized radio equipment may be necessary, and more reliable AC power or off-grid renewable energy resources will be required to ensure consistent operations.

Geography and Terrain

Where is data communication needed? Are there specific locations where network nodes are required? What level of RF coverage will be needed in order to reach those locations? The places that the network must reach will determine the number and position of AREDN® nodes.

What are the geographical characteristics of the area across which your data network will operate? Different types of terrain may require specific types of network connections in order

to adequately cover the region over which data communications are needed. More demanding terrain may require a larger number of intermediate nodes or possibly larger higher-gain antenna systems and mounting structures.

Expansion and Growth

Will your network need to expand or adapt to changing conditions over time? Mesh networks are ideally suited for *ad hoc* growth and least cost routing based on the availability of nodes. As more devices are added to the network, however, a simple *ad hoc* mesh topology will not properly scale in size. It could result in increased latency on the network, with some network segments becoming almost unusable if application response time thresholds are exceeded. A growing network will probably require a different well-designed topology that routes data traffic efficiently in order to reach its intended destination.

Applications and Throughput

What network programs, applications, or services should be provided in order to fulfill the purpose for this network? Each application will generate a certain amount of data traffic, and some programs or services are more data-intensive than others. The network needs to be designed to adequately pass the traffic for the required applications.

How many simultaneous users will be generating network traffic at different times? As the number of users increases, the amount of data traversing the network will also increase. In addition, with an increasing number of nodes on the network there will be a corresponding increase in the amount of routing traffic that is necessary to maintain the network. An AREDN® network should be designed to handle the expected workload.

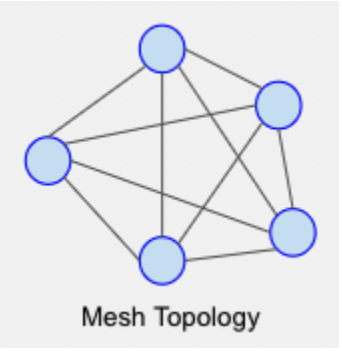
With these issues in mind, it is always best to keep your network as simple as possible and to include only those services which are required. Be sure to design your network so that it accomplishes its mission and suits its intended purpose.

CHAPTER

TWELVE

NETWORK TOPOLOGIES

Every AREDN® node is capable of automatically joining an *ad hoc* mesh network which is operating with the same SSID, channel, and bandwidth. New nodes will each explore their surroundings by broadcasting their identity and listening for their neighbors' responses. Once nodes identify others within radio range, they share this information so that each node has a picture of the network topology. Periodic updates adjust the network routes based on changes in signal quality or loss of a link, allowing the network to adapt to changing conditions. Since there can be several possible routes between nodes, and since network disruptions typically effect only part of the network, a mesh topology can provide redundancy for network links.



Mesh Topology

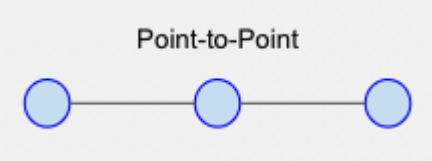
Every AREDN® node within radio range of other nodes will be able to participate in the network to extend its reach, provide route redundancy, or host services needed on the network at large. This simple mesh topology may serve its purpose perfectly for a short-term network deployed in support of a local event, or even for more permanent communication between nodes which are always within radio range. However, as mentioned in the previous chapter, the most important consideration for you network design is, *"What is the purpose for this particular network?"* The specific requirements of your mission should drive the design of your data network.

12.1 Types of Topologies

Although AREDN® nodes are capable of forming a simple mesh network, it is more common for operators to use different topologies in order to accomplish their data communication goals in growing networks. Typical network designs include Point-to-Point, Hub-and-Spoke, Tree or hybrid topologies.

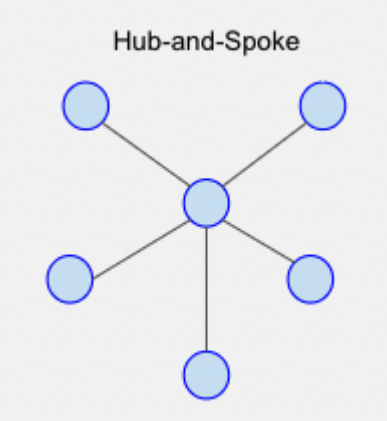
Point-to-Point Topology

Point-to-Point topologies are best suited for moving data between the far endpoints, potentially using one or more intermediate nodes in order to traverse different types of terrain or to overcome obstacles in the network path.



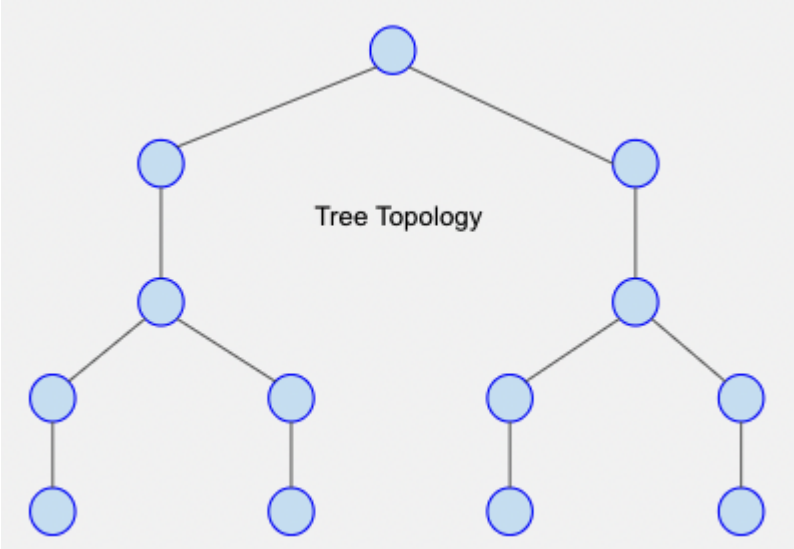
Hub-and-Spoke Topology

Hub-and-Spoke topologies work well in situations where the data communication to outlying nodes should be coordinated or funneled through a central location. Even if a remote node becomes unreachable, the rest of the network can continue to operate; but if the central node goes offline, the network will not function.



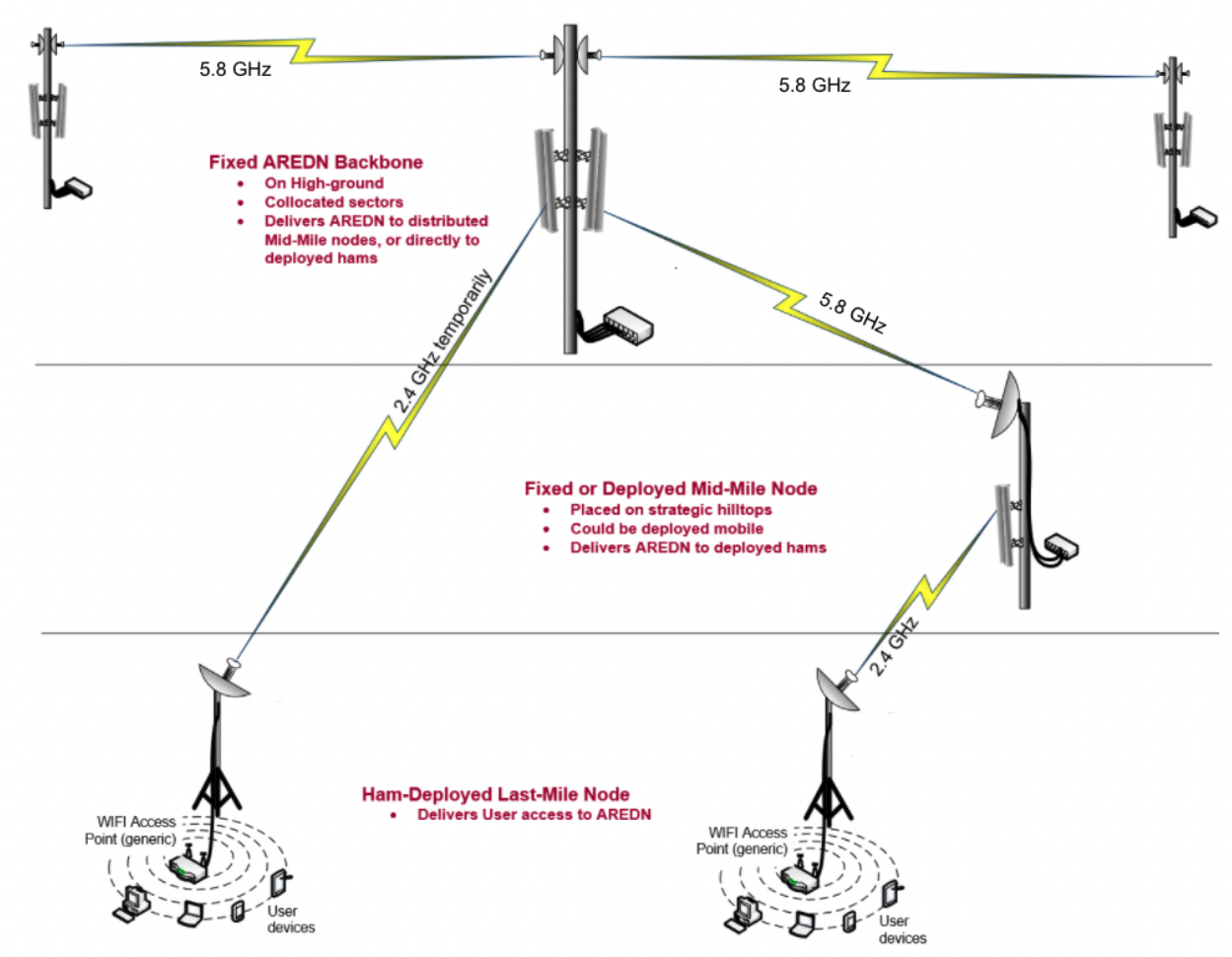
Tree Topology

A tree topology can be used to segment or partition network traffic, keeping specific data within a localized area while also allowing for links to remote parts of the network. The tree topology uses a parent-child hierarchy to structure the paths that data can take. This design can be easily scaled up or down to meet the specific requirements of the mission, but it does create "single points of failure". If nodes go offline within the hierarchy then entire branches of the tree can become unreachable.



12.2 Types of Links

A *link* consists of both sides of a radio path, including the two devices that communicate back and forth across that path. Depending on the specific goals and the RF environment, there may be a need for special types of network links that connect the areas where data communication is required to fulfill your mission.



Backbone Links

As the name implies, these links form the backbone or superhighway along which large amounts of data can travel for long distances at relatively high speed. Typically backbone or "backhaul" links are permanent installations on mountain peaks, tall buildings, or high towers. They are usually point-to-point links with large high-gain antenna systems running on reliable power sources. In some cases these links are designed with redundant radios which help ensure path protection. Backbone links can operate over distances between 10 to 30+ miles.

Relay Links

Relay links bridge the gaps between endpoint nodes. Their primary purpose is to pass data efficiently, but there may be cases where they also serve as network access points for users. Sometimes these links are called "mid-mile", "distribution", or "intermediate" nodes. They are usually installed on medium-height towers or buildings in order to achieve high signal quality with good line of sight to other relay or backbone nodes. Depending on conditions, intermediate links may operate over distances between 3 to 10+ miles.

Endpoint Links

Endpoint links are used to connect destination nodes to the network. Sometimes these links

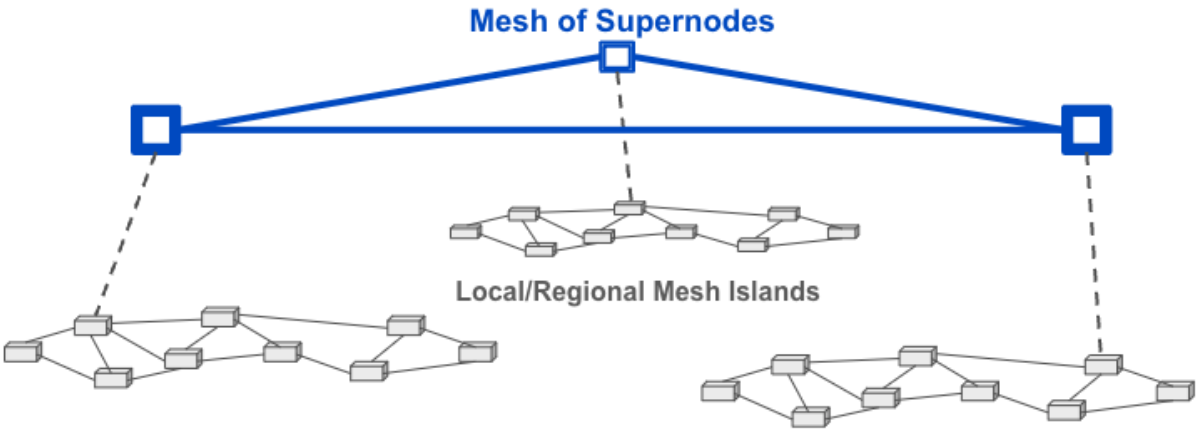
are called “last mile”, “tactical”, or “terminal” links. Usually the nodes at the far end will serve either as the originators or the final destinations for network traffic. Depending on local conditions, endpoint links typically operate over distances of 3 miles or less.

Different types of radio links may be needed to connect all of the nodes that are required in order to fulfill the purposes for your network. The ultimate goal of your network topology is to have a reliable data network that accomplishes its purpose for providing services to the intended destinations and users.

12.3 Supernode Architecture

Once several local or regional networks have been created, there may be a need for access between these “mesh islands.” In the past, node owners used direct Internet tunnel connections to accomplish this. However, this has the effect of merging the mesh islands into a single network with all of the routing traffic traversing all of the member networks. Many nodes were unable to handle the increased load.

A more efficient solution is to use the Supernode network to provide access across mesh islands, without sharing all of the local routing traffic across the linked networks. The Supernode network is a high-level mesh network — `super` meaning *“above or higher.”* The Supernode network sits above the isolated mesh networks and provides connectivity while insulating the local networks from the normal routing load.



Example Usage

If you need access to a service that is running on a remote network, click the Cloud mesh icon  to view services available across the Supernode network. You can use the search box at the top of the page to limit the display to a specific search string (such as a callsign or a service name). Once you have located the remote service you can click its link to open that service or node. This allows you to have full interaction with the remote service without requiring a dedicated network link between your mesh and the remote network.

A Supernode is a specialized node whose sole purpose is to link with other Supernodes and to shield each local network from the aggregate routing traffic. **Cudy TR3000** or **OpenWRT One** hardware is recommended for Supernodes, along with an Internet connection that provides robust bandwidth. It is also possible to implement a Supernode on a *Mikrotik hAP ac2/ac3* or a Virtual Machine running AREDN® firmware. For more information, refer to *Configuring a Supernode* in the **How-To** section of the documentation.

CHAPTER

THIRTEEN

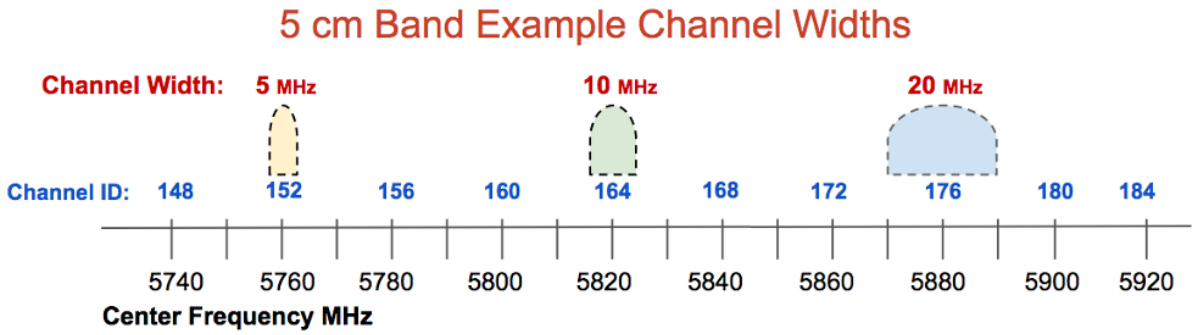
RADIO SPECTRUM CHARACTERISTICS

AREDN® networks operate in the microwave radio spectrum, and licensed Amateur radio operators have unique access to some of these frequencies. For bands in which Amateur operators share the spectrum, there is more chance for RF interference which may make some frequencies unusable for AREDN® data networking. For best results, select frequencies that are not being heavily used within the coverage area.

Warning: You are responsible for using frequencies, channels, bandwidths, and power levels that comply with your country's Amateur radio license requirements.

Channel Information

Each band is divided into channels, each of which consists of a 5 MHz frequency offset identified by the center frequency of the channel and assigned a numerical label. In the example below you can see that a selected channel may use more or less of the frequency range based on the chosen channel width. The wider the channel, the more overlap there will be with adjacent channels. Wide channels have the effect of reducing the number of non-overlapping or non-interfering channels that will be available for use. When selecting channels and widths, be sure to use non-overlapping channels. Devices using channels or channel widths that overlap will interfere with one another and cannot communicate to coordinate the sharing of bandwidth.



Some or all of the bands shown below are shared with other authorized users. For example, all of the upper channels on the 13 cm band are shared with standard FCC Part 15 WiFi (IEEE 802.11x)

users in the US. The following table shows examples of the Amateur radio bands, frequency ranges, and number of 5 MHz wide channels that are available for AREDN® networking in the US.

Band	Frequency Range	Channels
5 cm	5650-5925 MHz	54
9 cm	3300-3445 MHz	14
13 cm	2390-2450 MHz	10
33 cm	902-928 MHz	(varies)

The choice of a frequency band for AREDN® networking depends on several different factors, but you can "mix and match" bands in your network design as long as both sides of a radio link use the same band, channel, and channel width.

You have the option of selecting the channel width for each link. When using channels at the top or bottom of a band, be certain that your chosen width will not transmit outside of the FCC Part 97 allocation for that band. Different channel widths may yield better throughput than others. In some areas operators use different channels to isolate links, so they may need to use 10 MHz rather than 20 MHz channels in order to ensure they have enough available channels. Also, long distance links simply have better performance using 10 MHz vs. 20 MHz or 5 MHz channel widths. Test the performance of your links using various channel widths to ensure that they are optimized.

Power Limitations

The power limits that apply to AREDN® networks are the same as those that apply generally for Amateur radio operators in your country. As with any other operating mode, you should use the *minimum* power required to make radio links between nodes. In the United States, for example, this rule is specified in FCC part 97.313(a), and the maximum transmitter output power cannot exceed 1.5 kW PEP as specified by FCC part 97.313(b).

However there is one situation in the US where AREDN® devices are limited to 10W PEP. This special limitation applies to legacy devices that use 802.11b, which is a Spread Spectum (SS) emission. FCC part 97.313(j) limits SS transmitter power to 10W PEP. All other AREDN® devices use 802.11n which transmits carrier waves with combinations of PSK and AM modulations. Refer to the 802.11n MCS rate tables for specific modulations that are used.

In actual practice, the output power of AREDN® devices will be limited by the hardware that is used. Even though in the US the FCC rules allow higher power, all of the modern commercial routers being used for AREDN® physically cannot transmit these high power levels. Therefore, the power limits allowed in the US by the FCC will never be reached unless you have an external Power Amplifier.

Some of the advantages and disadvantages of each frequency range are explained in the sections below which give examples of frequencies that are available to Amateur radio operators in the US.

13.1 5.8 GHz Characteristics

Advantages

One advantage for using the 5 cm band is that it contains 54 channels, and many of them may be under-utilized with less chance of interference. You can choose channel widths of 5, 10, or 20 MHz, with larger channel widths providing higher data rates. Remember that reducing the channel width may increase the SNR (Signal to Noise Ratio) to improve signal quality if that is an issue for a marginal radio link.

The radio equipment and antenna systems for this band are readily available and are less expensive due to greater consumer demand. There is a wide variety of equipment from several manufacturers which supports the AREDN® firmware and operates across the 54 available channels. Radio and antenna systems for this band which are similar in size to those for other bands will often have higher gain. Devices in the 5 cm band are also well-suited for *Backbone Links* since there is little chance for RF interference from other radios sharing these frequencies at high profile sites. With clear line of sight and well-aligned antennas, 5.8 GHz signals can propagate across very long distances.

5.8 GHz																			
	Channel	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
	Ctr Freq	5.655	5.660	5.665	5.670	5.675	5.680	5.685	5.690	5.695	5.700	5.705	5.710	5.715	5.720	5.725	5.730	5.735	5.740
	Status	Shared with US unlicensed indoor/outdoor DFS & Radar Avoidance															Shared with Unlicensed...		

149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166
5.745	5.750	5.755	5.760	5.765	5.770	5.775	5.780	5.785	5.790	5.795	5.800	5.805	5.810	5.815	5.820	5.825	5.830
Shared with US unlicensed indoor/outdoor																	

167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184
5.835	5.840	5.845	5.850	5.855	5.860	5.865	5.870	5.875	5.880	5.885	5.890	5.895	5.900	5.905	5.910	5.915	5.920
...Shared with Unlicensed				Shared with US unlicensed mainly indoor									*Shared with Intelligent Transportation System*				

Disadvantages

One concern with all of these frequency bands is that there must be clear line of sight between the nodes on each side of the link. This means that not only do the nodes need to have an unobstructed direct path, but the Fresnel Zone between the nodes must also be clear. The diameter of the Fresnel Zone depends on the frequency and the distance between nodes. If less than 20% of the Fresnel Zone is obstructed there is little signal loss, but any blockage beyond 40% will cause significant signal loss and could make the path unusable. For a link in the 5 cm band with 10 miles between nodes the first Fresnel Zone radius will be 46 feet, which is much less than the frequency bands discussed below. However, the 60% no blockage radius in the 5 cm band is still about 28 feet. Be sure to account for node AGL (height Above Ground Level) and terrain in order to achieve clear line of sight between nodes.

13.2 3.4 GHz Characteristics

Attention: Late in 2020 the FCC ruled to sunset secondary Amateur allocations in the 9 cm *(3.3-3.5 GHz)* band. Although existing Amateur operations *"may continue while the Commission finalizes plans to reallocate spectrum,"* be aware that future FCC actions could remove Amateur operations altogether. Consider this before investing in or implementing new AREDN® devices in this band.

Advantages

Equipment in the 9 cm band is appropriate for *Backbone Links* since there is less potential for interference from other devices sharing these frequencies at tower sites. With clear line of sight and well-aligned antennas, 3.4 GHz signals can propagate across very long distances. You can select channel widths of 5, 10, or 20 MHz, with larger channel widths providing higher data rates. Remember that reducing the channel width may increase the SNR to improve signal quality if that is an issue for a marginal link.

3.4 GHz															
	Channel	76	77	78	79	80	81	82	83	84	85	86	87	88	89
	Ctr Freq	3.380	3.385	3.390	3.395	3.400	3.405	3.410	3.415	3.420	3.425	3.430	3.435	3.440	3.445
	Status	US Amateur operations remain on a secondary basis but are subject to removal at any time by FCC notice*													

* per FCC 20-138 IV-E-69

Disadvantages

Equipment for the 9 cm band is no longer being manufactured and used devices are becoming difficult to find. Care must be taken when selecting radios so as not to confuse them with the more common WiMAX devices which are designed for the 3.65 GHz range and are not supported for use with AREDN® firmware. As mentioned previously, there must be clear line of sight and the Fresnel Zone between nodes also must be clear. For a link in the 9 cm band with 10 miles between nodes the first Fresnel Zone radius will be 62 feet, which is less than the 13 cm band discussed below. However, the 60% no blockage radius is still about 37 feet. Consider node AGL and terrain in order to minimize obstructions.

13.3 2.4 GHz Characteristics

Advantages

One advantage for the 13 cm band is that radio equipment and antenna systems are more readily available and less costly due to higher consumer demand. There is a wide variety of equipment from several manufacturers which supports the AREDN® firmware and operates

in this band. With clear line of sight and well-aligned antennas, 2.4 GHz signals can propagate across very long distances.

Within the available frequency range you have the option of selecting channel widths of either 5, 10, or 20 MHz. A larger channel width will provide higher data rates. However, one effect of reducing the channel width is to increase the SNR to improve signal quality. For example, changing from a 20 MHz to a 10 MHz channel width will result in a 3 dB signal gain and could make the difference between a marginal link and a usable one. Just remember that when you cut the channel width in half you are also reducing your maximum throughput by half. Carefully test your links to ensure optimal performance.

2.4 GHz														
	Channel	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8 *
	Ctr Freq	2.387	2.392	2.397	2.402	2.407	2.412	2.417	2.422	2.427	2.432	2.437	2.442	2.447
	Status	non-US only		*Unshared*		Cannot Use	Shared with US unlicensed							

* Only 5 MHz channel width is available on channel 8

Disadvantages

The upper channels of the 13 cm band are shared with several other FCC authorized services. Depending on local RF conditions it may not be possible to use these shared channels because of the high noise floor which reduces SNR and decreases signal quality. This leaves licensed Amateur operators only two unshared channels with a possible bandwidth of 5 or 10 MHz each.

As mentioned previously, there must be clear line of sight and the Fresnel Zone between nodes also must be clear. For example, on a link in the 13 cm band with 10 miles between nodes, the first Fresnel Zone radius will be 72 feet. In the 13 cm band the 60% no blockage radius is approximately 43 feet, which is often higher than most *Intermediate* or *Last Mile* nodes have been installed. Careful consideration must be given to node height and the terrain between nodes in order to minimize obstructions.

13.4 900 MHz Characteristics

Advantages

The advantage of this band is that its longer wavelength may make it better suited for penetrating some types of foliage which would normally block signals at higher frequencies. Its NLOS (Non Line of Sight) propagation characteristics may be what is needed in order to establish an RF link between challenging locations. Legacy equipment for the 33 cm band provided only four 5 MHz wide channels (as shown below).

900 MHz	Channel	4	5	6	7
	Ctr Freq	907	912	917	922
	Status	Shared with US unlicensed			

Recent advances in wireless technology have introduced devices which use the Wifi HaLow (802.11ah) protocol. This protocol provides relatively high data rates while minimizing media contention, extending coverage range, and using low power levels. These devices may provide coverage in challenging areas. The number of usable channels varies based on the selected channel width (1, 2, 4, or 8 MHz). For example, at 1 MHz width there are 23 channels, while at 8 MHz width there are only three channels.

Disadvantages

The entire 33 cm band is shared between several FCC authorized radio services. The disadvantage of using this band for AREDN® networking is that the entire band is quite narrow (25 MHz), and in some areas the RF noise floor may be high which reduces the available SNR. Legacy equipment for the 33 cm band is no longer being manufactured and is becoming difficult to find. However, the newer HaLow devices provide significant advantages over the older 802.11agbn radios.

Different frequency ranges are available to connect the mesh nodes that are required in order to fulfill the purposes for your network. As you plan the frequencies to be deployed at specific locations, it may be helpful to use a *spectrum analyzer* for identifying ranges that are already in use. The ultimate goal is to have a reliable data network that accomplishes its purpose for providing services to the intended destinations and users.

CHAPTER

FOURTEEN

CHANNEL PLANNING

The previous section identified the different channels in each frequency band which are available for AREDN® networking. Devices on each side of a radio link must use the same frequency band, channel, channel width, and SSID. Beyond that requirement, however, you have quite a bit of flexibility to select the radio channels that will ensure the highest signal quality and throughput for your network. In a basic AREDN® network with several nodes spread across a limited geographical area, all of the nodes may use the same band, channel, and channel width. This allows them to establish network routing to any of the sites as needed.

However, as more nodes join the network or when several nodes are COLLOCATED (same physical site) and share the same channel, it is possible for overall network performance to degrade. In order for an AREDN® network to scale up in size and complexity, frequency coordination and channel planning become increasingly important. To plan for future growth, local AREDN® groups may need to partition use different network topologies and to allocate different channels for specific geographic areas or types of links in order to ensure the network will be able to support the expected services.

14.1 Wireless Network Operation

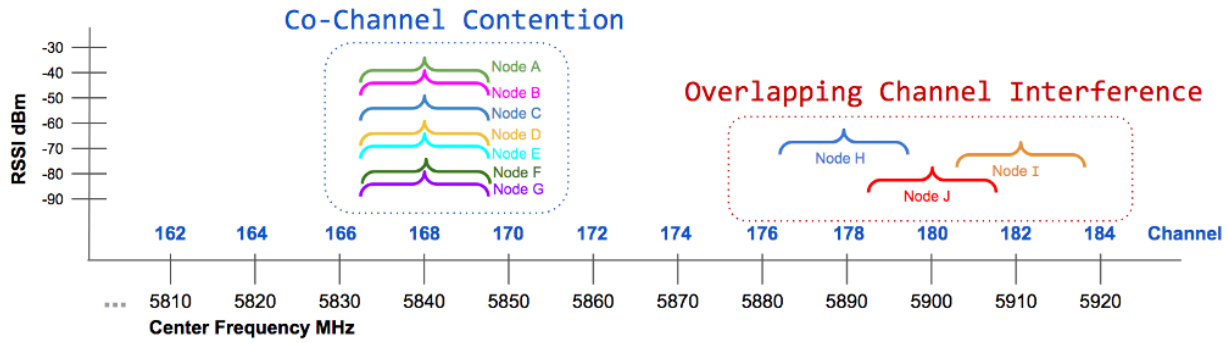
A wireless network is a shared half-duplex medium on which only one station at a time should transmit. In that sense wireless operations are analogous to other types of radio transmissions. If two stations key up their transmitters at the same time, they will interfere with each other to the extent that neither of them will receive the other's message. That is why net control procedures are implemented to ensure controlled access to a radio channel during emergency communication.

AREDN® firmware automatically mediates station access to the wireless medium by implementing IEEE 802.11a/b/g/n standards and Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA). This listen-before-talk technology helps nodes to determine whether a channel is busy. Each node performs a *Clear Channel Assessment (CCA)* as well as using Request to Send / Clear to Send (RTS/CTS) messages to negotiate access to a channel. A negligible amount of network traffic is also required for routing protocols to maintain routes for the network as a whole, but this traffic is a very small fraction of the total.

In a single-channel wireless network, any node that needs to transmit will automatically coordinate with the other nodes for a clear channel. This is by design, but as more devices try to gain access to the same channel there is an increased potential for each node to wait longer for its chance to transmit. This can result in increased latency and decreased network throughput as the number of network nodes increases.

14.1.1 Channel Contention

The concept of *Overlapping Channel Interference* is illustrated on the right side of the following channel scan diagram. *Overlapping Channel Interference* is very serious, but it can be eliminated by selecting non-overlapping channels for all of the devices accessing your network. A second issue related to how wireless networks operate is illustrated on the left side of the diagram. It is commonly called *Co-channel Interference* but is more accurately described as *Co-channel Contention* or *Co-channel Cooperation*.



In this example several nodes must share a single channel, so they all negotiate for the opportunity to transmit. Any node that needs to transmit will use listen-before-talk technology to determine whether the medium is busy. If the channel seems clear, the node will attempt to transmit data. If the channel is busy, the node will defer transmission until the channel seems clear. In a high-density network where a large number of nodes share a single channel, the normal negotiation processes may result in significant performance degradation. From an end-user perspective, an overloaded channel can make the network seem sluggish or even unusable.

This example is not meant to show that having only seven nodes will overload a channel. There is no established rule of thumb for channel sharing that specifies how many nodes are too many. The answer depends on the number of nodes, the bandwidth in use to support required services, the link signal qualities, and other network characteristics. Based on these parameters one shared channel may perform well with many dozens of nodes, while another network may see performance degradation with significantly fewer nodes sharing a channel. Many factors interact to influence network performance, but it will soon become obvious to users whether the network is behaving as expected.

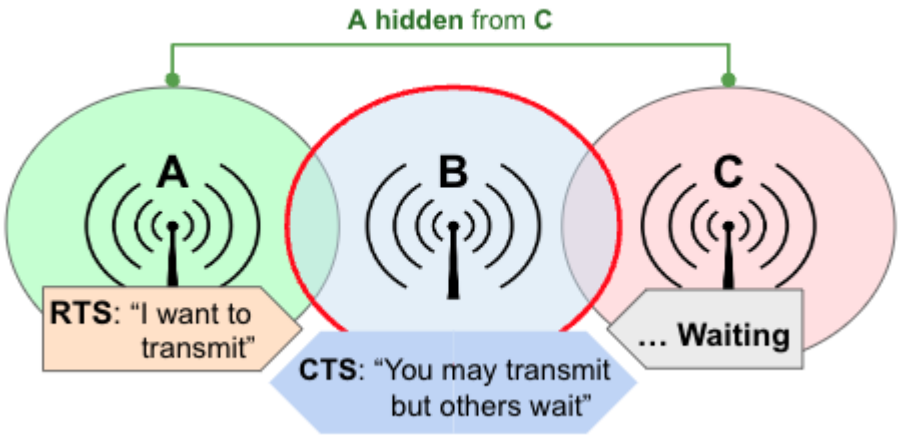
Several tools are available for testing network performance such as *ping* to measure latency, *traceroute* to identify how traffic is being routed, and *iperf3* to estimate network throughput. Periodic

measurements along with user perceptions can be helpful in determining whether channel separation would be of benefit. It is an expected by-product of how wireless networks normally operate, but performance can be enhanced by planning the assigned channels for your mesh devices as described in the **Channel Plans** section below.

14.1.2 Hidden and Exposed Nodes

Hidden Nodes

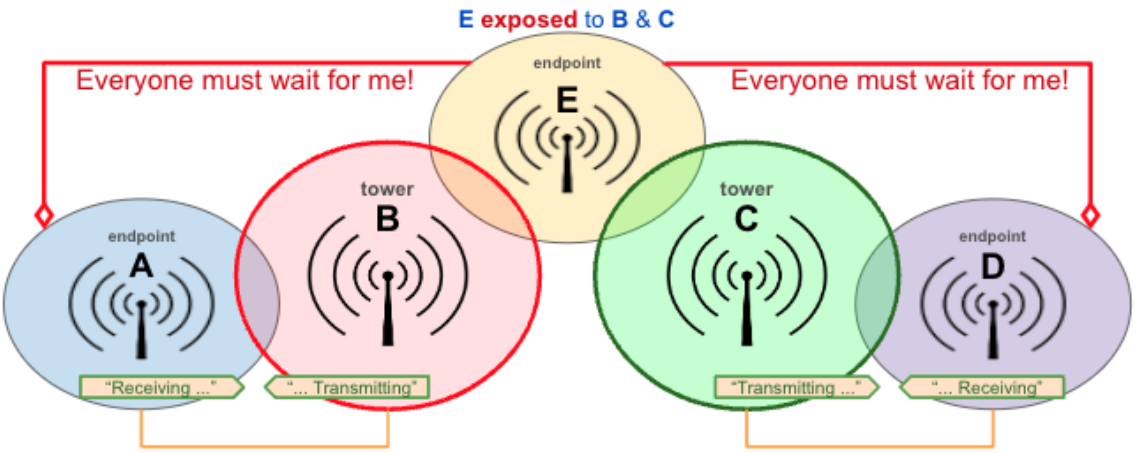
In any wireless network there will be nodes that are not within radio range of each other but which share the same channel. In the Hidden Node example below, node **A** can reach node **B** but cannot reach node **C**. Since **A** and **C** are hidden from each other, they may try to transmit on the shared channel at the same time without knowing it. Because of their relative locations and any associated network delays, each node may think it has a clear channel for transmitting.



Request to Send / Clear to Send (RTS/CTS) messages can be used by AREDN® nodes to minimize these issues. For example, node **A** broadcasts a short RTS message with a proposed timeslot/duration for transmitting its data stream. Node **B** receives that request and broadcasts a CTS for that time slot. Node **C** could not hear the original RTS but will hear the CTS message and defer its transmissions during that time slot.

Exposed Nodes

In the Exposed Node example below, Endpoint **A** and tower **B** can communicate with each other at the same time that tower **C** can communicate with endpoint **D**. However, if endpoint **E** is exposed to *both* of the towers, then the tower nodes will detect that the channel is not clear and will not be able to communicate when the exposed node is transmitting. This increases the network wait time which impacts overall throughput.

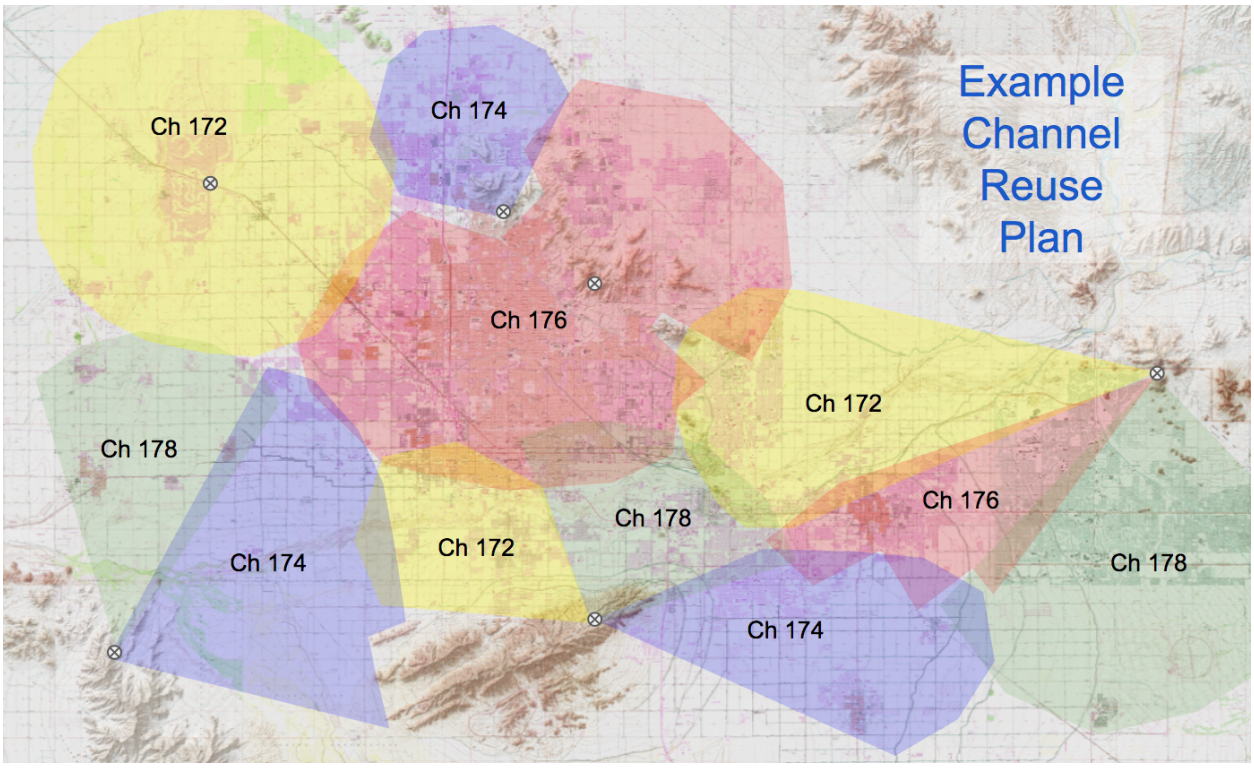


Try to eliminate the exposed node problem by placing them onto different bands or channels along with the nodes you want them to communicate with. Since nodes using directional antennas are nearly invisible to others not positioned in the antenna's beam, directional antennas should be used with care when sharing a channel so that exposed nodes are not created unintentionally. If you have exposed nodes that are causing throughput degradation, segment each group of nodes by putting them on different bands or channels.

14.2 Channel Plans and Frequency Coordination

You may experience poor network performance if there are too many nodes using the same band and channel. Here is a simple example to illustrate the issue: a three-hop path from QTH1 to Tower1 to Tower2 to QTH2. If all links are using the same channel, then only one node at a time can send data. This instantly cuts the throughput by one-third or more and increases latency with protocol overhead. To improve performance you could configure each link to use a different channel, allowing simultaneous transmissions. For example, the collocated tower nodes could be DtD linked via Ethernet, with QTH1 and Tower1 using 5 GHz channel 172 while QTH2 and Tower2 use channel 176. Before this channel plan is implemented it might be possible to have one HD video stream and one VoIP call with frequent dropouts. After the channel plan is implemented it should be possible to have three HD video streams and several VoIP calls simultaneously with few dropouts.

Depending on the frequency band you are using, there are varying options available for assigning non-overlapping channels to your mesh devices. For example, in the 5.8 GHz band using even-numbered 10 MHz channels, there are 25 non-overlapping channels. Ideally, RF coverage zones (sometimes called "cells") should use different channels. Overlapping cell coverage can provide broader connectivity, but the overlapping coverage zones should not use overlapping RF frequencies.



The example coverage map shows that four different channels have been assigned to achieve broad coverage by segmenting specific areas into zones to reduce co-channel contention. It should be noted that even a channel reuse plan such as this may not eliminate all instances of contention. For example, if a node is at the outer edges of a coverage zone or is elevated well above ground level, its transmissions may propagate into a distant cell using the same channel. The radios in the other cell will defer if they hear the original node's transmissions, even though they originate in a different cell. Some degree of experimentation may be required in order to minimize contention and maximize network throughput.

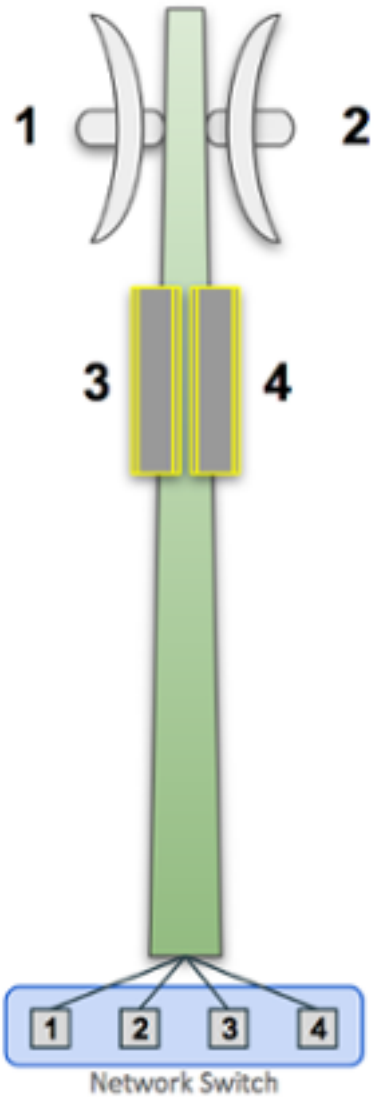
14.3 Collocated Nodes

At some sites there may be several devices mounted on the same building or structure. This photo shows many nodes collocated on a mountaintop.

Network performance degradation can occur if these nodes share an RF band and channel. For example, when two sector antennas are collocated and share the same channel, the network throughput for that site will be reduced by half or more. If you have collocated nodes then it makes sense to allow the devices to pass traffic over their Ethernet interface (as described below) rather than forcing them to use their radio channel.

14.3.1 Device to Device (DtD) Linking

In its most basic configuration for two collocated nodes, an Ethernet cable is connected between the PoE *LAN* port of each device. The routing protocol will assign a very low "link cost" to the DtD connection and automatically route traffic between the nodes over Ethernet rather than using the RF channel.

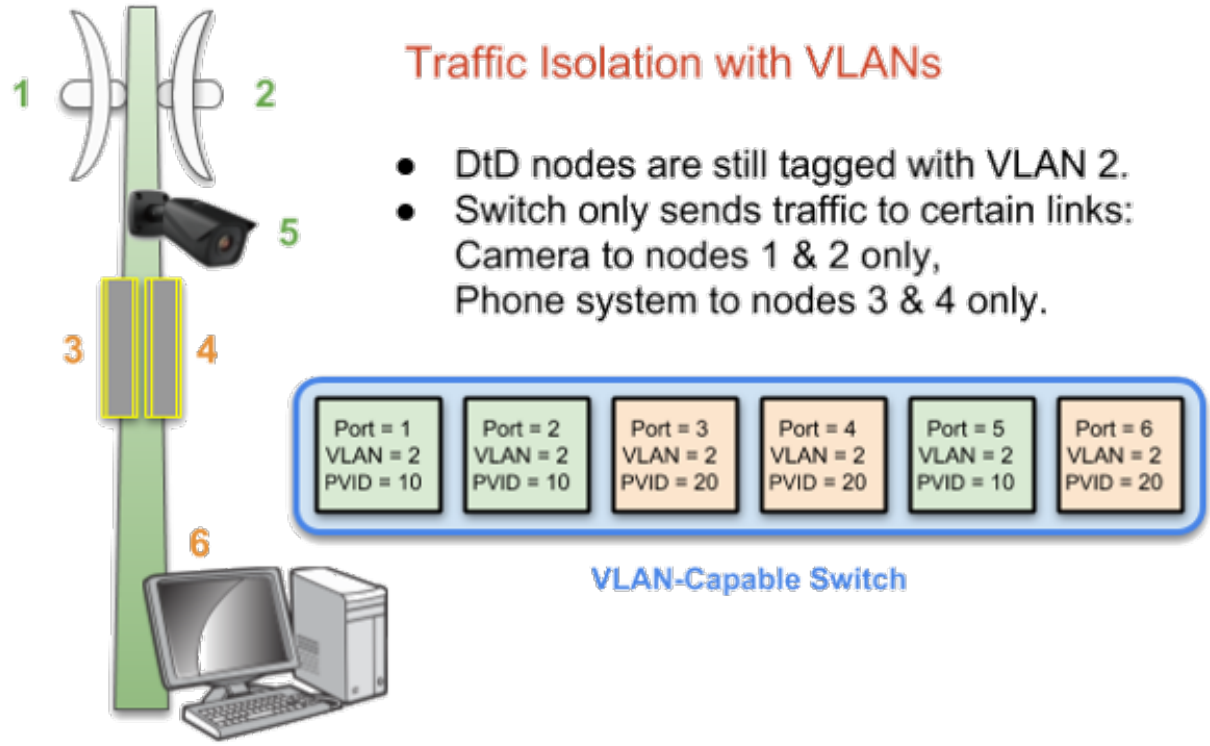


One added benefit of DtD linking is that you can link nodes which are operating on different bands and channels. Nodes that are using *Channel Separation* to segment traffic can still pass data at high speeds through their DtD link and be members of a single network. At a tower site like the one shown here, you could link 2.4 GHz and 5.8 GHz nodes to the same network. In fact, at a busy site like this it is best practice to use DtD linking, because otherwise RF channel contention could make the network unusable.

Ideally you should configure your collocated nodes to use different bands and channels, then set up DtD links between the nodes to ensure that traffic is routed efficiently without generating RF contention or delays. The routing protocol should always choose the DtD path first when passing traffic between linked nodes. Each AREDN® node recognizes incoming packets tagged with VLAN (Virtual Local Area Network) 2 as DtD traffic. In the simple example shown here, the switch will share all traffic across all ports and every node will receive the traffic on its DtD link.

Be aware that if several nodes are connected through a network switch (as shown in the diagrams) and then you connect your laptop to an open port on that switch, your laptop may receive a DHCP IP

address from any of the nodes' DHCP servers. This may not be an issue for a laptop doing periodic maintenance activities at the site. However, if you deploy another device which must receive a consistent DHCP IP address, then it is best practice to disable the DHCP server on all but one of the nodes which will be the primary DHCP server for any local devices connected to that network switch.



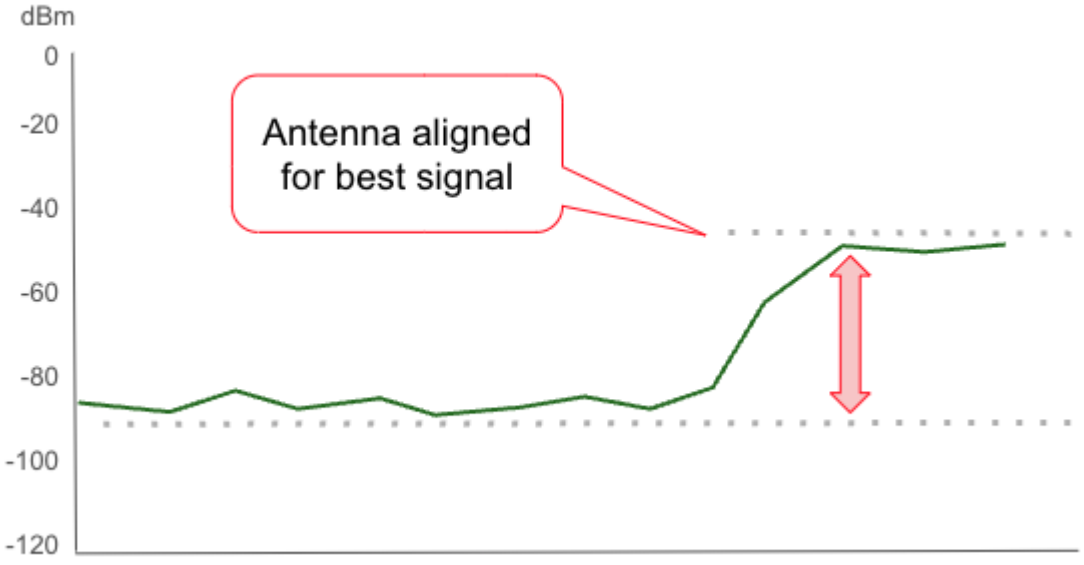
If you want to partition traffic even further, you can configure VLANs on a managed switch to isolate port traffic so that only the nodes which should receive specific traffic will see it. For example, you may have a video surveillance system (5) or a VoIP PBX system (6), and traffic from those devices should only be passed to a specific set of links as shown in the diagram below. The port-based VLANs will ensure that traffic is controlled and routed, rather than being broadcast across every link.

14.3.2 Antenna Polarization

Most of the latest AREDN® devices use dual polarity antennas and MIMO (Multiple Input - Multiple Output) features in the radios that exploit multipath propagation. However, if you are using single polarity antennas with "single chain" radios, another way to achieve signal separation for collocated devices is to orient the site's antennas so that one is vertically polarized and the other is horizontally polarized. This can result in a signal separation of up to 20 dB. Because of the predominance of vertical polarization in commercial WiFi devices, single chain AREDN® nodes may achieve slightly better performance using horizontal polarization with clear line of sight. You can test both polarizations to see which one yields better performance dealing with the man-made noise in your specific environment. Note that the antennas on both sides of a radio link must be oriented the same way.

14.3.3 Aligning Linked Nodes

The AREDN® web interface provides information that is helpful when aligning two nodes that are being installed to form a link. On the **Node Status** page, click the **Charts** button to view the *Realtime Signal to Noise* graph. Slowly turn and tilt your antenna, pausing to view the signal metrics. Once you see the best signal, as shown below, you can lock your antenna into position. If you want to focus on the antenna position without having to watch the SNR graph, you can also enable the *SNR Sound* feature and align the antenna to the highest pitch tone. Depending on the implementation, a Signal to Noise Ratio of 15 dB is adequate to pass data at speeds in the range of 5 to 20 Mbps (Megabits per second). See "Tips for Aiming Directional Antennas" in the **How-To Guides** section for additional information.



14.4 Channel Planning Tips

Network Scalability Tip

If there are two towers or cell coverage areas within range of each other, configure the nodes with different channels to avoid poor performance.

Based on the purpose for your network, try to create reliable paths to the locations where data is needed. Use channel separation and DtD linking of collocated nodes to avoid RF channel contention.

- If you need broad local coverage for a high profile area you can install sector antennas on a tower site: for example, three panels with 120 degree beam width each. DtD link the sectors at the tower site, and use different channels for each sector to avoid channel contention.
- Consider putting each local coverage area on its own channel to minimize the interaction between zones. Be sure to allow adequate RF separation between zones where channels are being reused.
- If you are installing long distance point-to-point links to connect network islands, be sure to use a separate band or channel for the backbone link. This type of link has a single purpose: to carry as much data as quickly as possible from one end to the other. Eliminate any type of channel contention so that these links can achieve high throughput.
- Remember that a multi-hop path through the network must have good signal quality on each leg of the journey. You cannot expect adequate performance through a series of poor quality links. For example, if you traverse three links having LQ (Link Quality) metrics of 65%, 45%, and 58%, your aggregate LQ will be 17% which is unusable. Ideally the aggregate LQ should be at least 80% to have a link that supports the applications and services you require.

CHAPTER

FIFTEEN

NETWORK MODELING

As you design your AREDN® network it is often helpful to estimate ahead of time whether a node or link might accomplish your goals for the network. One way to get this information is to use computer modeling programs that predict the performance of RF devices. There are many types of computerized tools that you can use, ranging from relatively expensive commercial software to freely available open source programs. You should select and become familiar with the tool that best fits your aptitude, experience, and budget.

In this section some free tools will be used to illustrate how to determine your network's available paths and overall coverage. Keep in mind that a computer modeling tool only provides a prediction and does not guarantee that two sites will be able to communicate when actually deployed.

15.1 Creating a Path Profile

Path profiles are very helpful for determining whether a link between two nodes will have clear line of sight and acceptable signal levels. In order to create a path profile you will need to have the following information for both of your node endpoints:

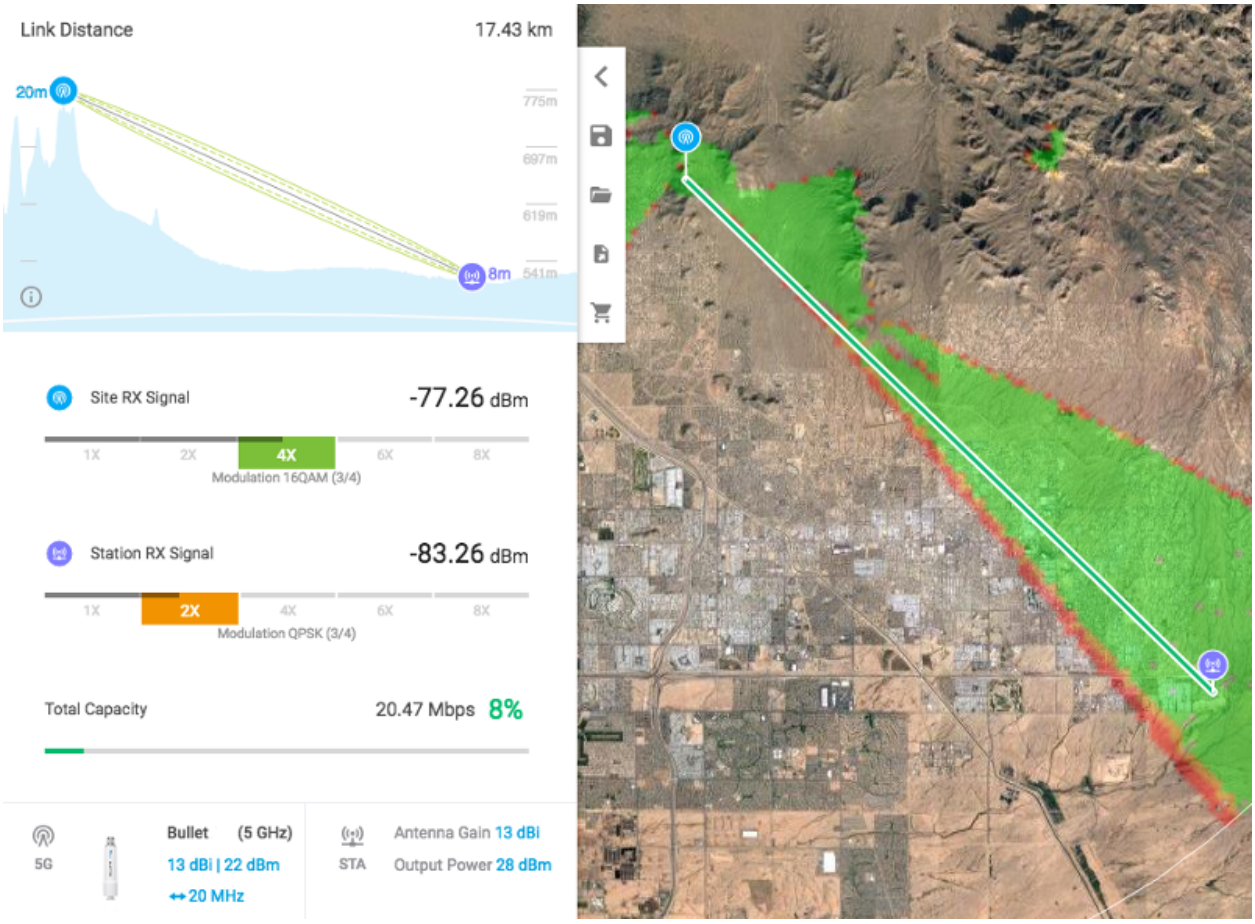
- Latitude and Longitude
- Antenna AGL
- Frequency
- Transmit Power
- Line Loss
- Antenna Gain
- Receiver Sensitivity

Most computer modeling software will be able to estimate the link characteristics given this information.

15.1.1 Ubiquiti AirLink Tool

If you are using Ubiquiti radios there is a free modeling tool available on the Ubiquiti website (https://ispdesign.ui.com). This tool will ask you to locate your node endpoints by clicking on a map display. It allows you to select the radio frequency and model from a dropdown list, as well as having you specify the antenna heights, antenna gain, and transmit power. With this information it will calculate and display the coverage area and the link quality.

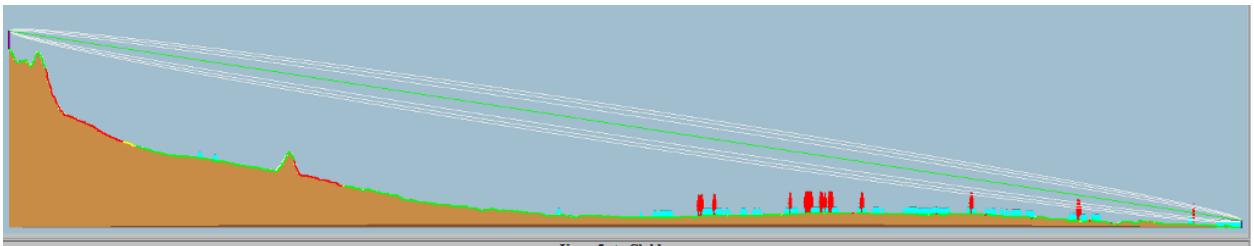
The path profile is color coded to indicate whether the link quality is adequate. It displays the link distance, line of sight, as well as the Fresnel Zone and 60% clearance area. It also estimates the signal levels at each endpoint and the predicted throughput for the link. An example *AirLink* path profile is shown below.



15.1.2 VE2DBE's Radio Mobile Tool

Whether or not you are using Ubiquiti devices, you can create detailed path profiles using VE2DBE's *Radio Mobile* software. This program is available for download, but it is very easy to use the web-based version: http://www.ve2dbe.com/rmonline.html

With *Radio Mobile* you must first create a *Site* for each of your endpoints. Then you can select the endpoints from your *Site* dropdown to generate a path profile between any of the listed locations. Once you enter the radio and antenna information in the link display, *Radio Mobile* will create your path profile. There are several metrics displayed here which may not be available in the Ubiquiti tool, including free space path loss, obstruction loss, forest loss, urban loss, and fade margin. This additional information may help you determine why a path is not working, and it may assist you with choosing alternate sites for node locations. Typically a fade margin of 15 dB or greater is adequate for a usable link. An example *Radio Mobile* path profile is shown below.



Usery 5g to Clubhouse

Usery Pass Tower		(2) Clubhouse	
Latitude	33.492655 °	Latitude	33.383602 °
Longitude	-111.640556 °	Longitude	-111.506051 °
Ground elevation	755.6 m	Ground elevation	512.0 m
Antenna height	25.0 m	Antenna height	10.0 m
Azimuth	134.14 TN | 124.09 MG °	Azimuth	314.21 TN | 304.22 MG °
Tilt	-0.93 °	Tilt	0.77 °

Radio system		**Propagation**	
TX power	28.45 dBm	Free space loss	132.52 dB
TX line loss	0.50 dB	Obstuction loss	-3.16 dB
TX antenna gain	20.00 dBi	Forest loss	0.00 dB
RX antenna gain	16.00 dBi	Urban loss	1.00 dB
RX line loss	0.50 dB	Statistical loss	6.50 dB
RX sensitivity	-94.96 dBm	Total path loss	136.86 dB

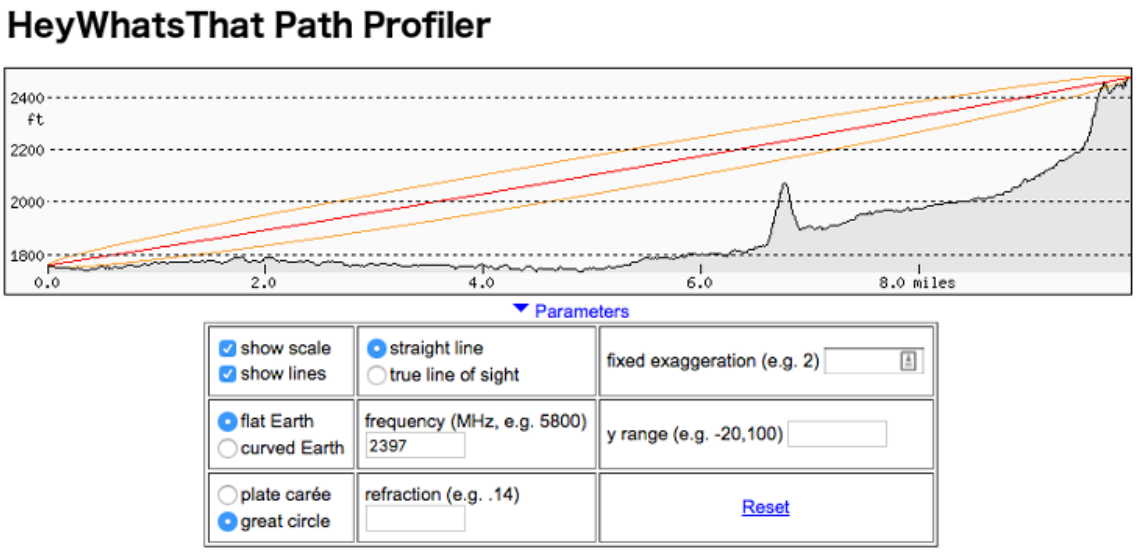
Performance	
Distance	**17.401 km**
Precision	10.0 m
Frequency	5825.000 MHz
Equivalent Isotropically Radiated Power	62.388 W
System gain	158.41 dB
Required reliability	70.000 %
Received Signal	-73.41 dBm
Received Signal	47.82 µV
Fade Margin	**21.55 dB**

15.1.3 HeyWhatsThat Path Profiler

Another web-based tool will generate a path profile from points selected on a map. HeyWhatsThat Path Profiler is available here: http://heywhatsthat.com/profiler.html

Simply click on the map at the bottom of the webpage to add an endpoint for each side of your link. Once an endpoint has been added, it can be moved by clicking and holding the endpoint while dragging it to the new location on the map. After adding your endpoints you will see the path profile displayed at the top of the webpage. You can click the *Parameters* link under the path display to specify additional items for the path calculation. If you specify the frequency then the Fresnel zone

for the path will be added to the display.



15.1.4 Radio Fresnel Tool

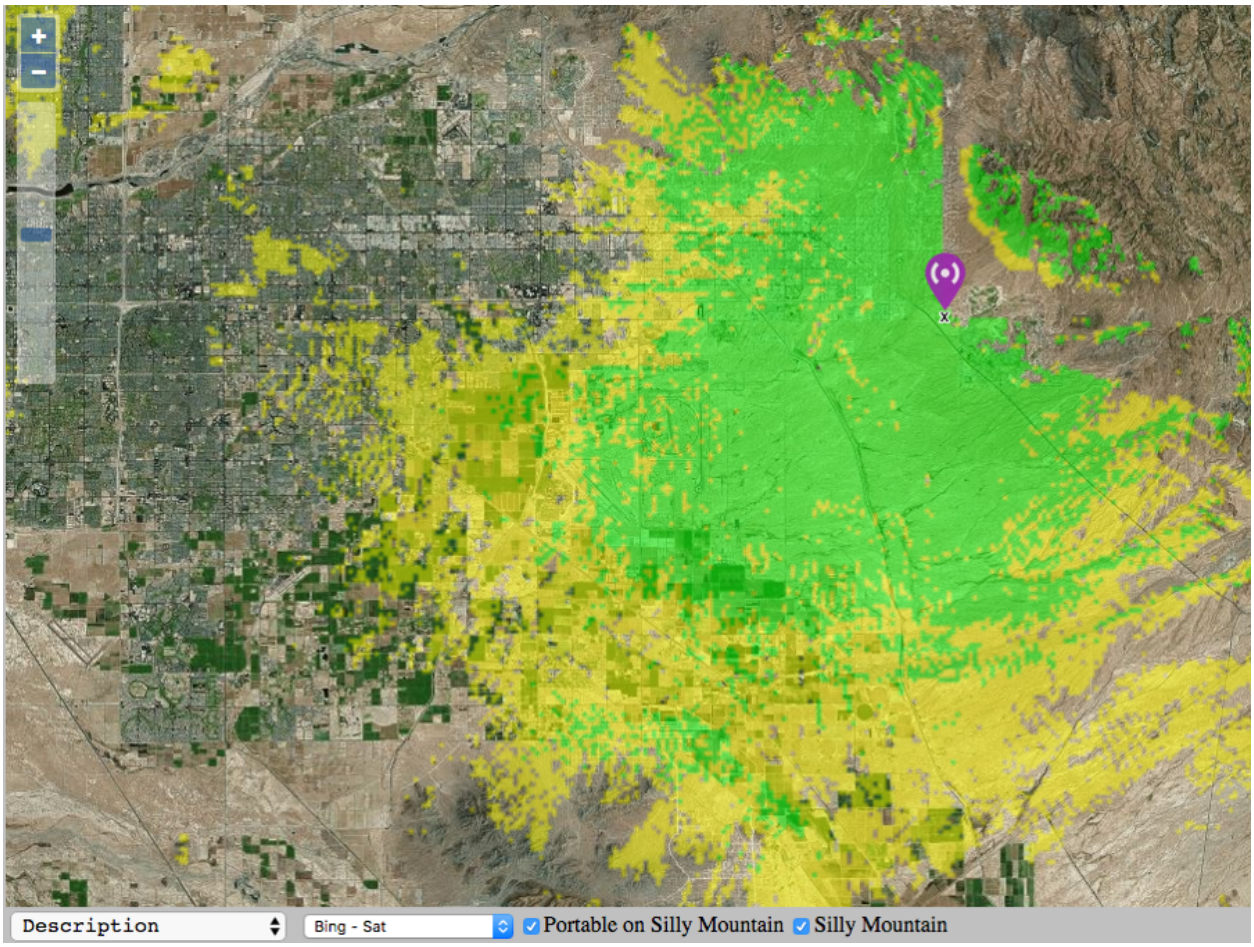
This web-based tool will generate a KML file which can be viewed as a 3D path profile using *Google Earth* software. Radio Fresnel is available here: http://www.radiofresnel.com

Simply enter the required site information into the online form and click the *Get KML* button at the bottom of the webpage. There is a sample KML file as well as a video tutorial for how to use the tool.

15.2 Determining Node or Network Coverage

In many cases it would be helpful to know ahead of time what area could potentially be covered with the signal generated by a particular node. Creating a coverage plot will show the predicted coverage on any of several types of base map.

An example *Radio Mobile* coverage plot is shown below. After entering the site, radio, and antenna characteristics the software produces a color coded map that predicts the areas of best, marginal, or no signal. One useful feature of *Radio Mobile* allows you to overlay several site coverage plots onto a single map so you can see the extent of coverage provided by multiple nodes in your network. Coverage maps such as these can show you the areas of adequate signal, as well as the "holes" which you may need to fill if you require more comprehensive coverage.


Description
Bing - Sat
Portable on Silly Mountain
Silly Mountain

CHAPTER

SIXTEEN

AREDN® SERVICES OVERVIEW

As mentioned in the AREDN® overview, the purpose of an amateur radio emergency data network is to provide typical Internet or intranet programs to people who need to communicate across a wide area during an emergency or community event. An AREDN® network provides the transport mechanism for the types of programs people typically use today to communicate with each other in the normal course of their business and social interactions. This may include keyboard-to-keyboard chat, email messages with images and attachments, file transfer, collaborative document sharing, VoIP phone service, video conferencing, GPS (Global Positioning System) tracking, surveillance camera streaming, computer aided dispatch, deployed resource management, weather station reporting, sensor monitoring and control, repeater linking, and many other services.

The purpose for this section of the AREDN® documentation is to identify examples of services that might be useful for communication across a mesh network. None of these programs are directly supported by the AREDN® development team. Almost any program that can operate on a peer-to-peer TCP/IP network is a candidate for AREDN® networking, but you should carefully select and test your services to ensure they will work within the following guidelines.

- An important consideration for selecting programs is to understand the impact each service will have on the performance and reliability of the network during the times when digital communication is required. As a best practice, choose programs which require the least amount of computing and network resources in order to operate successfully.

Note: The consideration above is especially important if you are deploying a service which regularly queries other nodes across the network. For example, if you deploy a network management system which polls metrics from remote mesh nodes, you need to carefully consider how many metrics you poll and how often you request them. Realize that polling dozens of metrics from each node every few seconds is likely to degrade mesh performance. Be sure to let node owners know what you are planning to do and get their permission/agreement for your polling schedule.

- It is equally important to choose data services that meet the criteria defined in FCC Part 97 regulations for amateur radio services. Try to avoid programs that use encryption or proprietary compression algorithms, which may be interpreted as “encoding messages for the purpose of obscuring their meaning” (FCC Part 97.113-a-4).

- As a general rule services should be run on separate LAN-connected computers rather than on the AREDN® nodes themselves. Node devices have very limited resources which should be conserved for node operation rather than for running extra programs. Try to select external computers that have low power requirements, since many AREDN® deployments are off-grid and without any external network access. Many operators use Raspberry Pi computers which are small, easy to transport, and require minimal DC power for operation.

When choosing programs to use as AREDN® services you will probably find that there is more than one way to accomplish your goals. It is crucial to clearly understand the types of communication that meet the requirements of your mission, and then you will be able to select the best programs for the job. Always try to use a program that will cause the least performance impact to your network.

Most TCP/IP programs are designed to use the Client-Server model, where one or more client programs communicate through a central server or servers distributed hierarchically. These types of programs can operate on a mesh network as long as the server is reachable or readily accessible by the nodes that need to use them.

As a general rule for mesh networks, simpler is better. The more complicated and automated you make your service design, the more network and computing resources will be required to operate the system. It is always best to conserve mesh networking resources wherever possible.

Several programs have been designed to take advantage of multiple paths between nodes and multiple peer servers coexisting on a mesh network. There are fewer of these mesh-friendly programs, but they will be identified as they appear in the following sections.

The remaining parts of this guide will focus on examples of services that could be offered on your AREDN® network. Programs are grouped by type, and where possible the network impact of each program will be described in order for you to understand the resources that may be required to use the program as a service on the mesh. Remember that the mentioned programs are merely suggestions or examples of typical Internet-style TCP/IP applications which could be deployed on you network to meet the specific communication requirements of your mission.

CHAPTER

SEVENTEEN

CHAT PROGRAMS

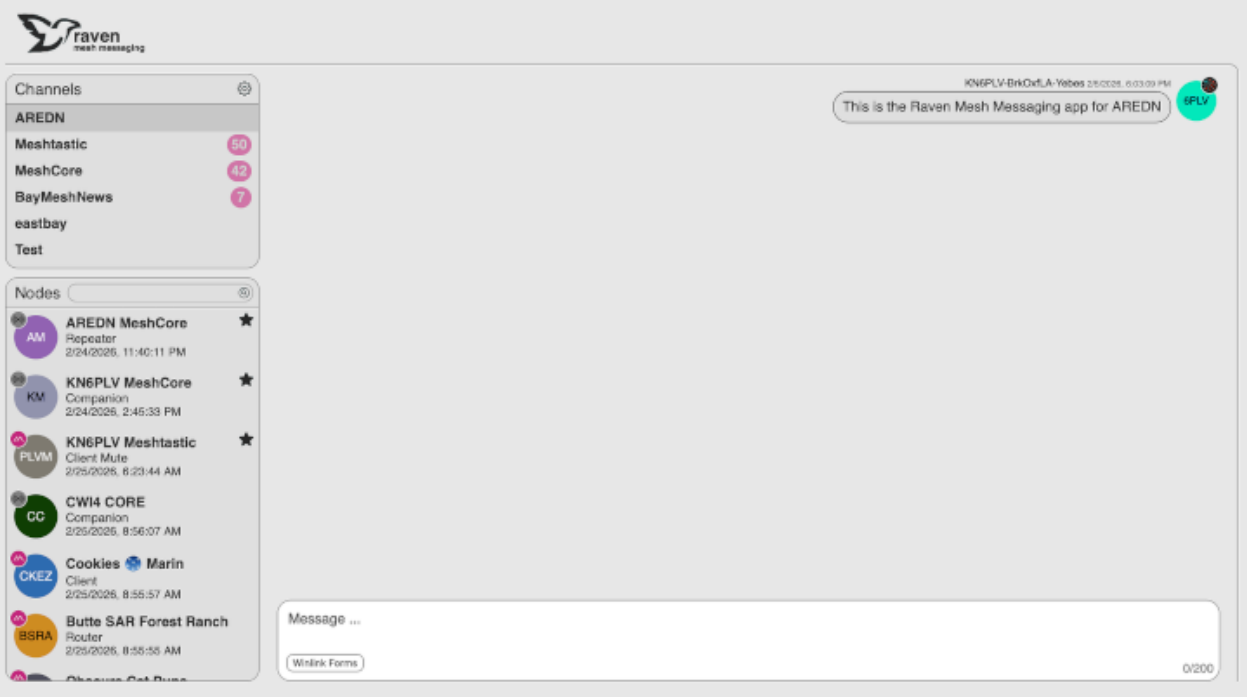
Online chat software includes any program which transmits short text messages between the sender and receiver. These realtime keyboard-to-keyboard messages create an environment similar to a spoken conversation. A chat session may involve one-to-one communication or group meetings. These programs are valuable for quick question/answer interactions where immediate replies are important. Timestamped conversation history is typically saved for future reference.

Chat programs are one of the least network-intensive types of communication programs, so they are a good candidate as low impact services on a mesh network. Many chat programs also offer file sharing, which allows you to get two functions within a single program. The following list is not comprehensive or complete but represents a sample of the types of chat programs that might be available for you to use as services on your mesh network. Only programs with open source licenses were included in this list, although commercial chat software can also be used.

17.1 Raven

Raven is a decentralized messaging platform with the ability to bridge messages between AREDN® and other message platforms. To participate in Raven it is recommended that you install the Raven package on your primary node, making sure that the node has plenty of available memory. Login to your node to launch the Raven app from the left navigation bar.

Channels maintain message streams separately from each other, and by default there are channels for *AREDN*, *Meshtastic*, and *Meshcore*. You can create additional channels for specific topics or purposes using the configuration display. Visit this site for additional information about Raven, and the Raven installation package (*apk* or *ipk*) can be downloaded from this site.



17.2 MeshChat

MeshChat is a popular chat service for AREDN® networks because it was written specifically for mesh communication by Trevor Paskett K7FPV. Users access MeshChat via web browser, and the service can run on the mesh node itself or on a LAN-connected Debian or Raspberry Pi computer. After logging in by entering a call sign, you can send a message by typing into a text box and clicking the *Submit* button. The list of active users is displayed, and every message is visible to all participants on the chat service. Multiple *Zones* and *Channels* are supported for categorizing and filtering message traffic.

A copy of the message database is stored on every device where MeshChat is running. Nodes may have intermittent network connectivity, but as long as at least one node is available the MeshChat database remains intact. Once nodes come online they immediately sync by retrieving a full copy of the message database. If any new messages are found, they are integrated into the local message database.

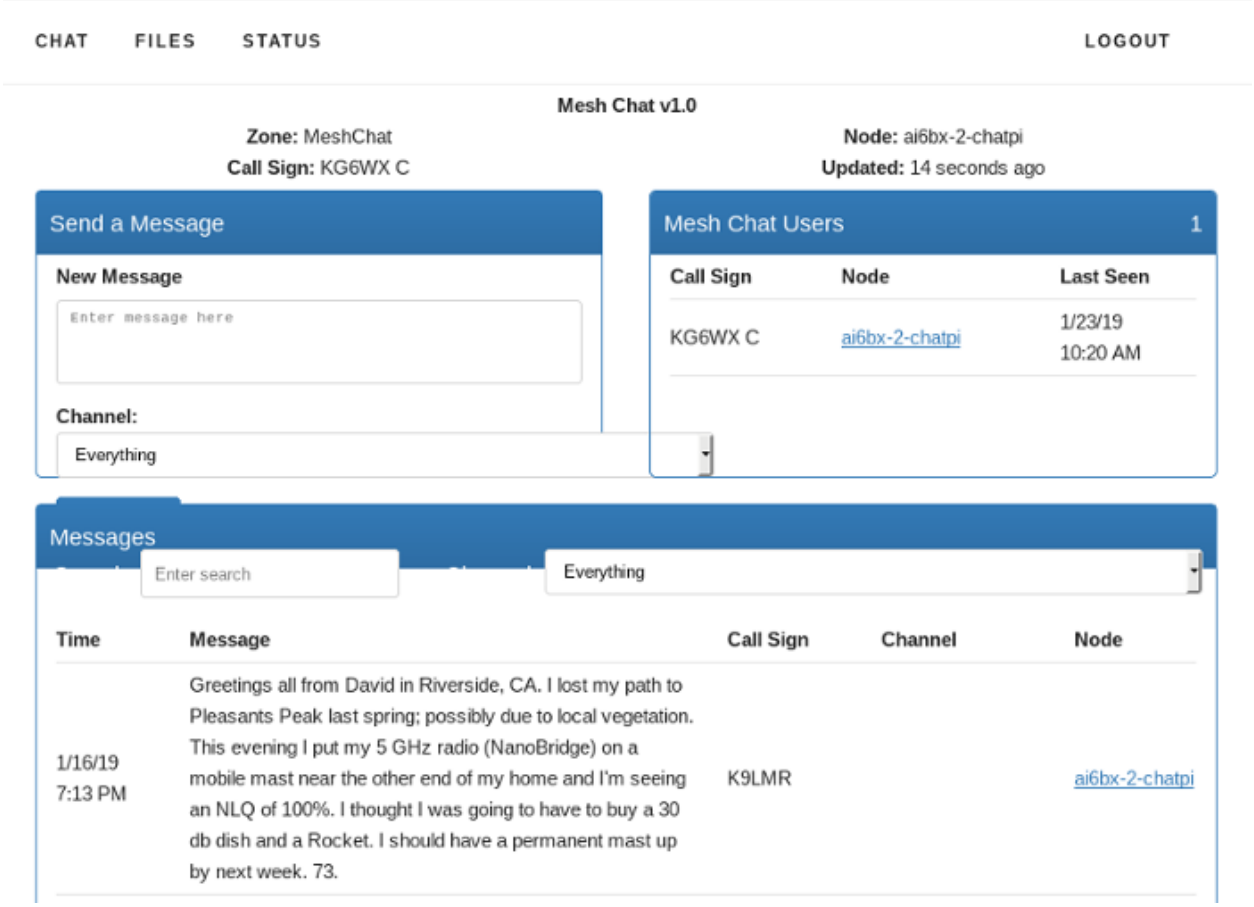
In addition to the keyboard-to-keyboard chat feature, MeshChat also allows files to be shared between nodes. Files may be uploaded from or downloaded to the user's computer using the web interface. If MeshChat is running on a radio node then the file storage is limited, but if running on an external LAN-connected computer the available storage for files is usually much larger.

MeshChat *Action Scripts* also provide for functional extensions, such as sending messages to an SMS gateway for external distribution. It is also possible for action scripts to periodically save the message database for archive purposes or integration with external tools.

Although MeshChat is a commonly deployed service, it is a third party package which is not

available in the AREDN® repositories. You can find additional information by visiting this link: MeshChat at Trevor's Bench

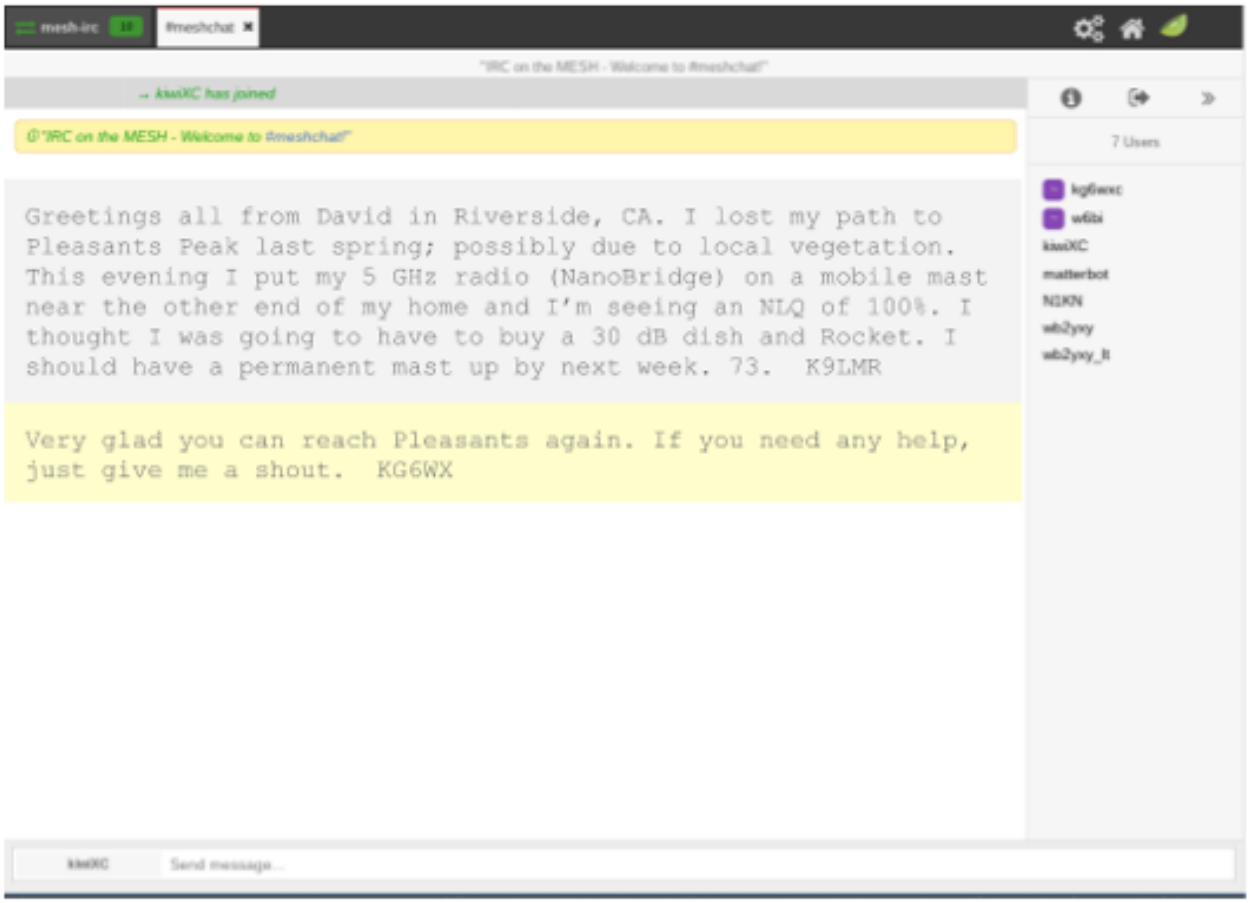
As originally designed, MeshChat was written in the Perl programming language. After the retirement of Perl on AREDN® nodes, an alternative was created using the Lua programming language. If you are running the original Perl version on an external computer, you can install the Lua API on your node to provide the computer with the list of MeshChat nodes. This package is available here: Lua version of MeshChat



17.3 Internet Relay Chat

Several implementations of Internet Relay Chat are available, either as open source software or in proprietary versions. The Internet Relay Chat Daemon (IRCd) is a server program that listens for connections from IRC client programs and brokers the communication between the connected clients. With this client-server architecture, the IRC server must be available on a network link with sufficient bandwidth in order for the clients to function.

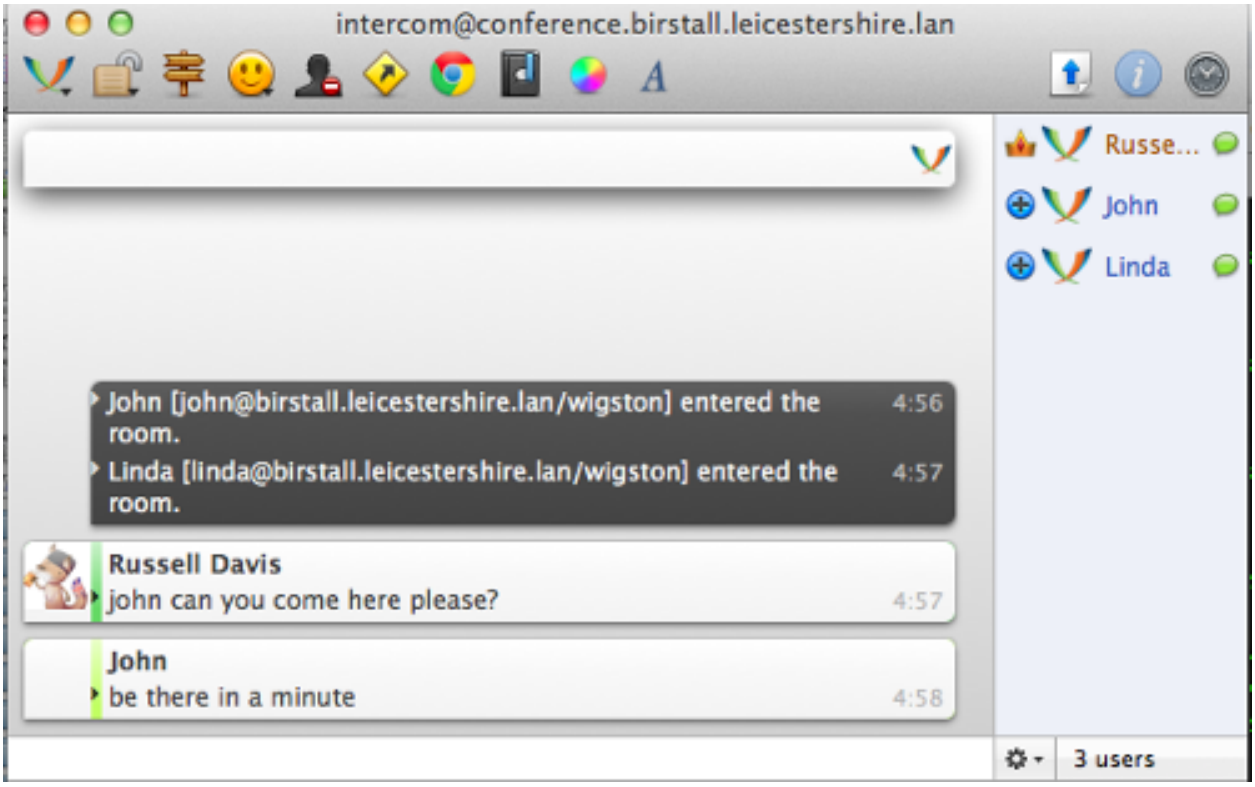
A wide variety of features and functions are available with these and similar chat programs, including various zones, channel types, and user roles. For additional information about IRC services, visit IRC Clients



17.4 Jabber/XMPP

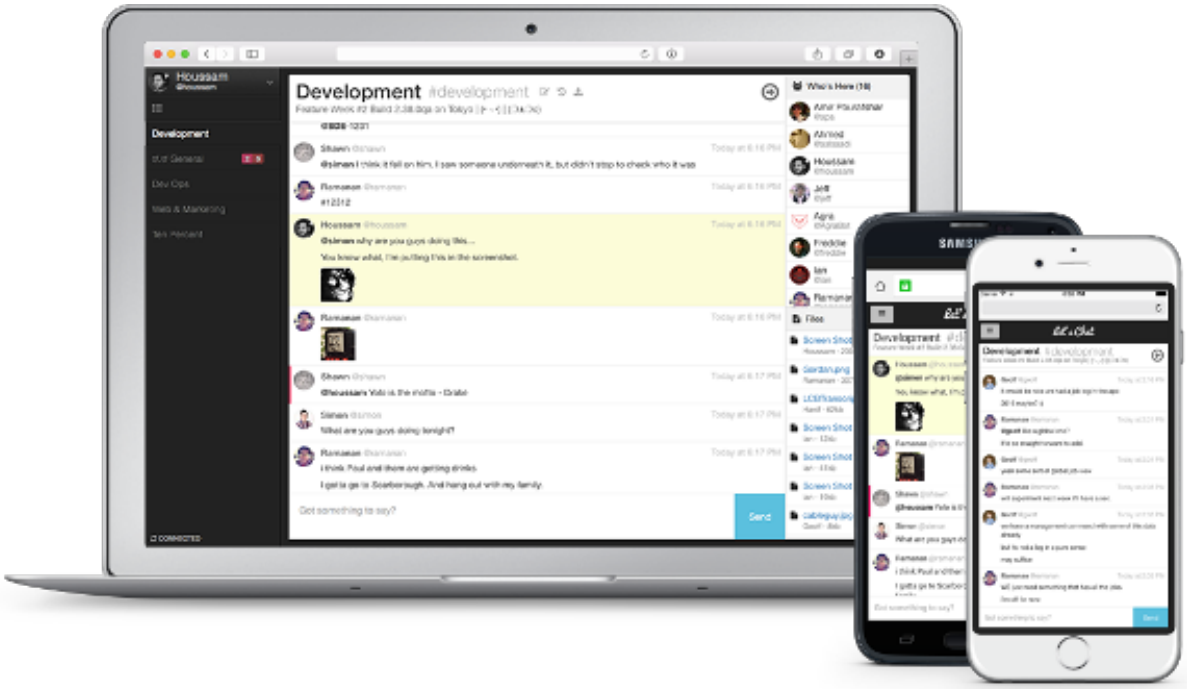
Originally known as Jabber, XMPP servers have been around for a long time but are fully compliant with modern messaging standards thanks to a large community of developers worldwide. These servers provide one-to-one messaging as well as group chat sessions. User lists have activity and presence indicators, and chat history can be archived for later use. There are dozens of feature modules available for XMPP servers which can extend the functionality as needed.

Two of the most popular XMPP servers are eJabberd and Prosody, but there are many others. For additional information about these services, visit the following links: eJabberd and Prosody



17.5 Let's Chat

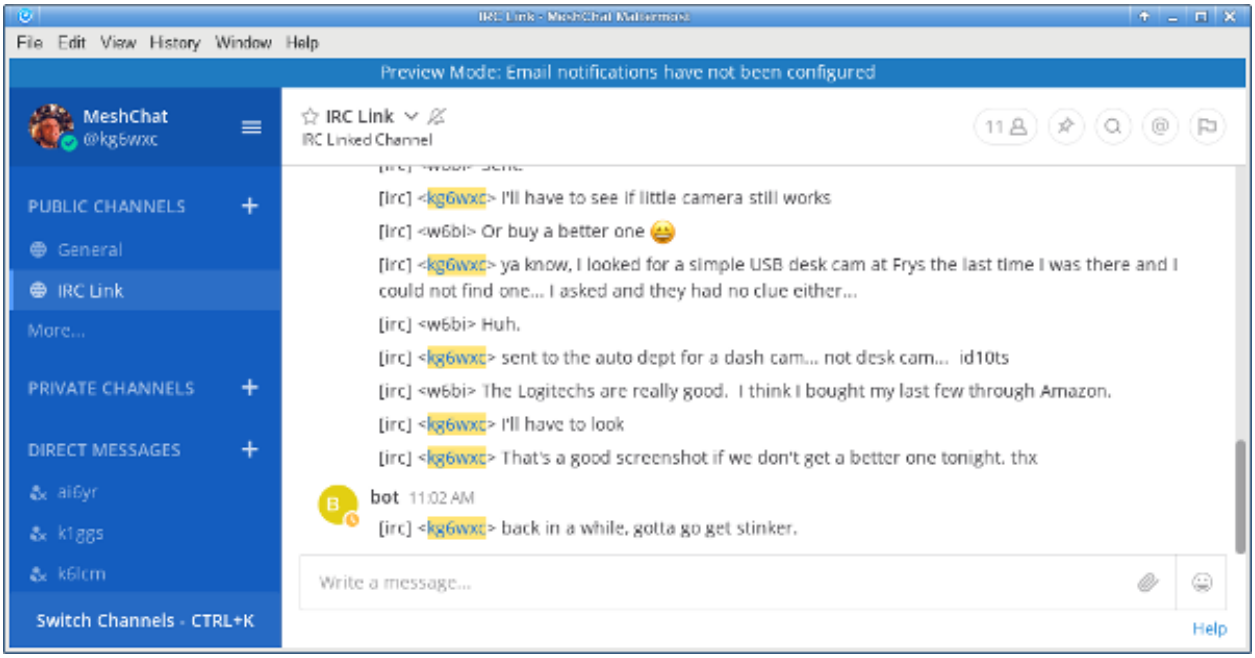
Let's Chat is an open source messaging service for small teams. It provides one-to-one communication between XMPP users as well as group messaging and @mentions in a variety of chat rooms. Searchable conversation history is available, in addition to text and image pasting, user activity notifications, and file uploads. User self-registration is configurable on the server. For additional information about Let's Chat, visit this link: Let's Chat



17.6 Mattermost

The *Mattermost Team Edition* is an open source platform that supports mobile and desktop messaging apps. It provides one-to-one and group messaging, file sharing, and message history with search capabilities. It is often described as an open source alternative to the commercial *Slack* communication tool.

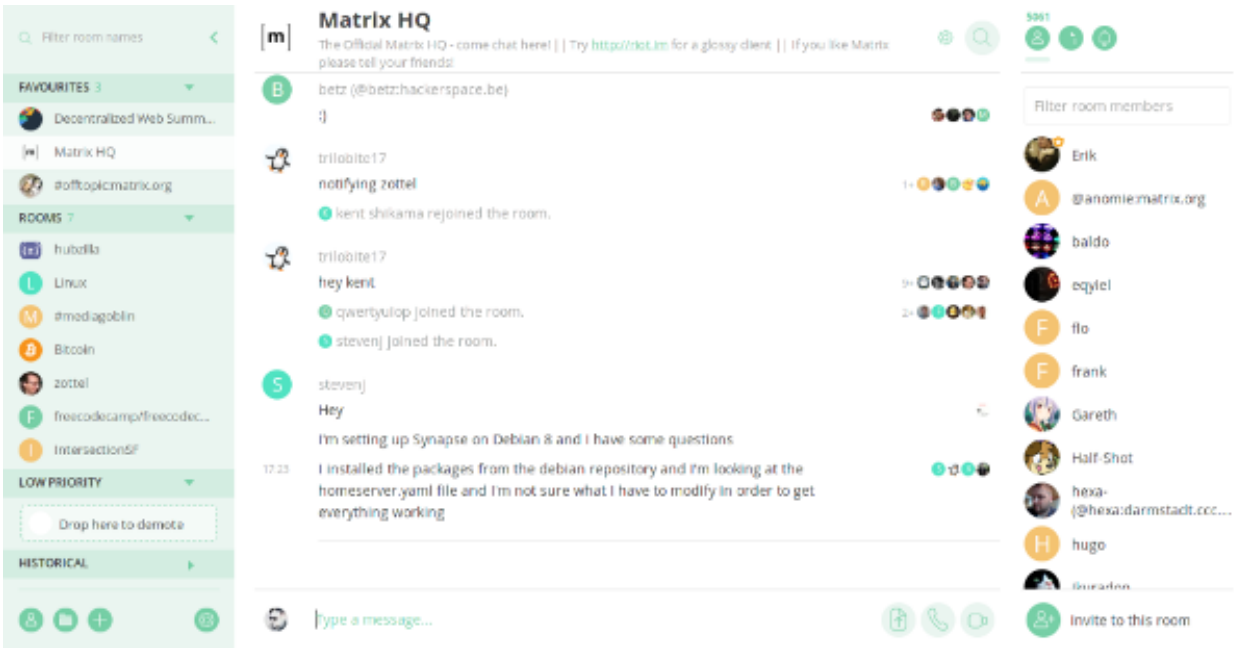
Mattermost supports @mentions, and channels are available for organizing conversations which can be topic-based, group-based, or event-based. Notifications indicate user presence and activity. File sharing is provided for PDF and text files, as well as audio, video, and image files. For additional information about Mattermost, visit this link: Mattermost



17.7 Matrix - Synapse

Synapse is the "homeserver" implementation of the *Matrix* communication platform. As with a traditional client-server architecture, every user runs a Matrix client that connects to a Synapse server which stores the personal chat history and user account information. However, these servers communicate with each other on the network, which creates a distributed content architecture that minimizes single points of failure.

Matrix services can provide one-to-one communication channels as well as group chats in a variety of rooms. User presence and typing notifications are supported, as well as chat history and read receipts. Although the Matrix platform is intended to provide end-to-end encryption, it can be run without cryptographic signing. Matrix can also integrate with IRC (Internet Relay Chat) services, as well as VoIP and video conferencing solutions via WebRTC. For additional information about Matrix-Synapse, visit these links: Matrix Home and Synapse



17.8 Example Chat Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	Architecture	Network Load	Age	Platform	Effort
Raven	distributed	small	new	node	easy
MeshChat	distributed	small	new	node	easy
IRCd server	client-server	small	old	lin/mac/rpi/win	medium
Jabber/XMPP	client-server	small	old	lin/mac/rpi/win	medium
Let's Chat	client-server	small	new	lin/mac/rpi/win	medium
Mattermost	client-server	medium	new	linux	expert
Matrix	distributed	medium	new	linux/mac	expert

CHAPTER

EIGHTEEN

EMAIL PROGRAMS

Email programs have become a communication standard for workers everywhere today. Email messages can include a wide range of information, from short chat-like interactions to lengthy and extensive text with complex document and image attachments. Whereas chat programs often assume that the sender and receiver are online at the same time, email programs use a store and forward approach to ensure message delivery even when users are not connected simultaneously.

Email operates on a client-server model. Users create or read their messages on some type of client program, although this software could be hosted on a network web server and accessed through a user's web browser rather than requiring a standalone email program to be installed on the client computer. Client programs typically access messages from the email server using either Internet Message Access Protocol (IMAP) or Post Office Protocol (POP). Client programs use Simple Mail Transfer Protocol (SMTP) to send messages to email servers, while the servers themselves use SMTP for both sending and receiving.

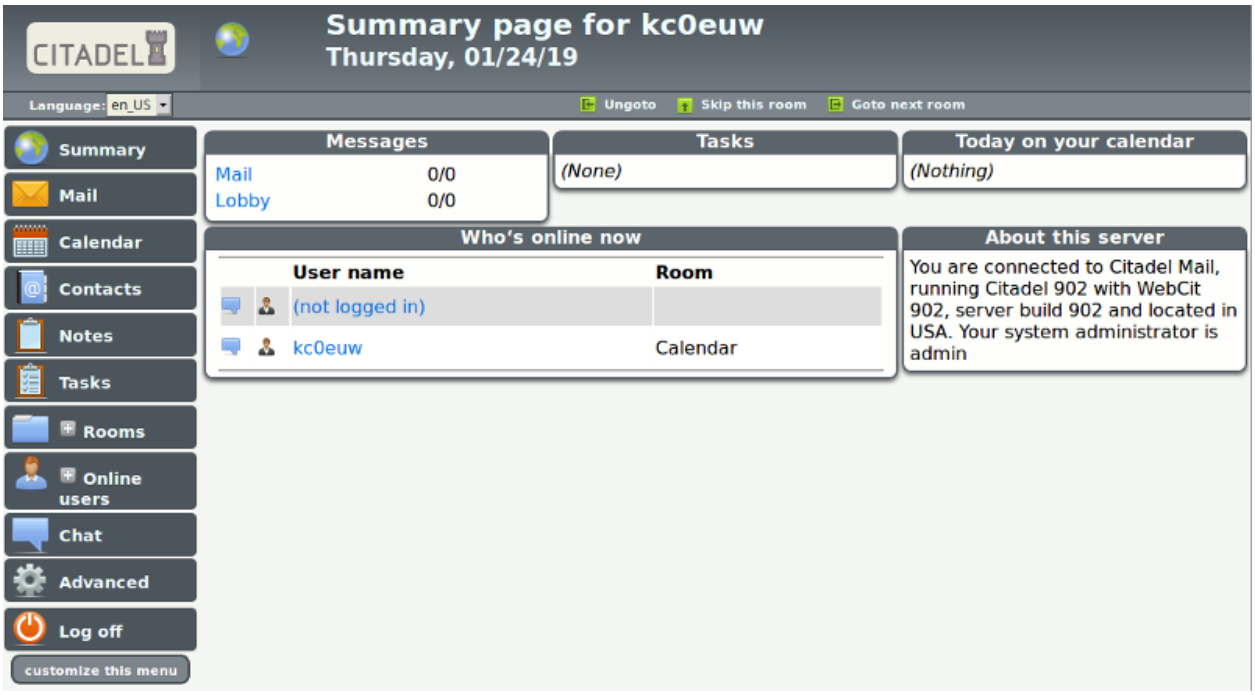
As with any client-server program, the email server must be reachable on a network segment with adequate bandwidth in order for the clients to exchange messages. If you have a choice, put your email server on one of your largest and most reliable network segments. Refer to this link for a comparison of email Client Programs, and visit this link for a comparison of email Server Programs. The following list is not comprehensive or complete but represents a sample of the types of software that may be available for you to use as services on your mesh network. With one exception, only programs with open source licenses were included in this list, although proprietary email software can also be used.

18.1 Citadel/UX

Not only does Citadel provide email, but it is also a full-featured *groupware* suite with chat rooms, calendars and scheduling, contact address book, file sharing, forum posting, and many other features. It contains built-in implementations of the following server protocols: IMAP, POP3, SMTP, XMPP, and ManageSieve. Citadel also provides user self-registration, which minimizes the administrative overhead of managing email addresses on the server.

Since a variety of features are bundled into a single application suite, Citadel is a less compli-

cated and more integrated way to implement several network services at once by installing a single package capable of running on a lightweight Raspberry Pi computer if necessary. Citadel's email services can be accessed using its browser-based webmail interface or from a separate email client program on a remote computer. For additional information about Citadel, visit this link: Citadel



18.2 Open Source Email Server

In order to implement an open source email server you will need to install several individual software packages, each of which will process one or more of the required email protocols. This is slightly more complicated than implementing a single groupware package such as the *Citadel* program described in the previous section. Protocols and example packages are described in the following lists.

SMTP

In order to implement an email server you will need to select a software package to handle the Simple Mail Transfer Protocol. You can select one of the example open source packages from the list below, or you can implement another SMTP agent of your choice.

- Sendmail is the original legacy SMTP server that is still used today, although one of the newer programs below is often chosen for its ease of configuration and added security features.
- Exim is the default SMTP server in Debian Linux, is well-documented, having many configurable features, and it runs from a single executable program.
- Postfix is the default SMTP server in Ubuntu Linux and MacOS, with many integra-

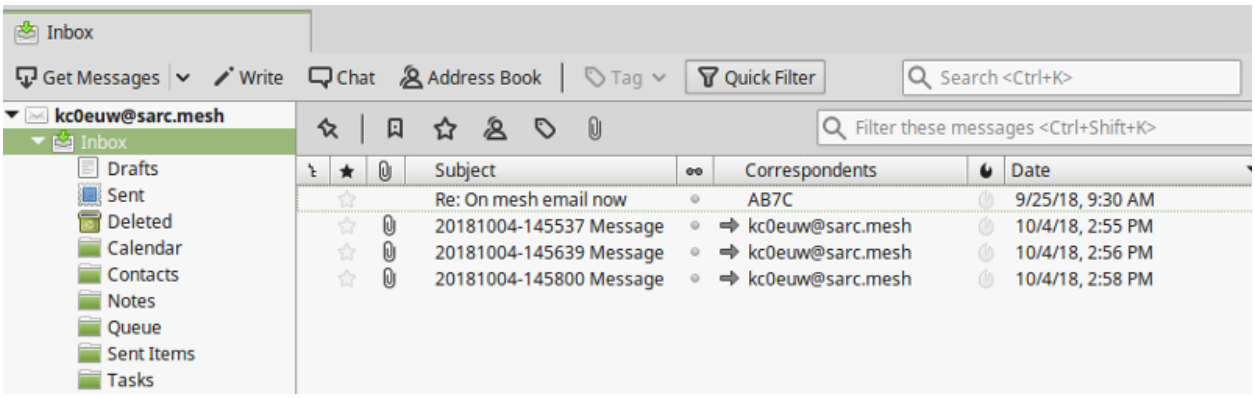
tion and security features, and it runs a series of parallelized programs for improved performance.

IMAP and POP3

In order for email clients to retrieve their messages you will need to select a software package to handle IMAP and POP3 communication. You can select the example open source package below or you can implement another IMAP/POP3 package of your choice.

- Dovecot is one of the most popular IMAP and POP3 servers for open source email systems, being found on more than 2/3 of the email servers across the Internet.

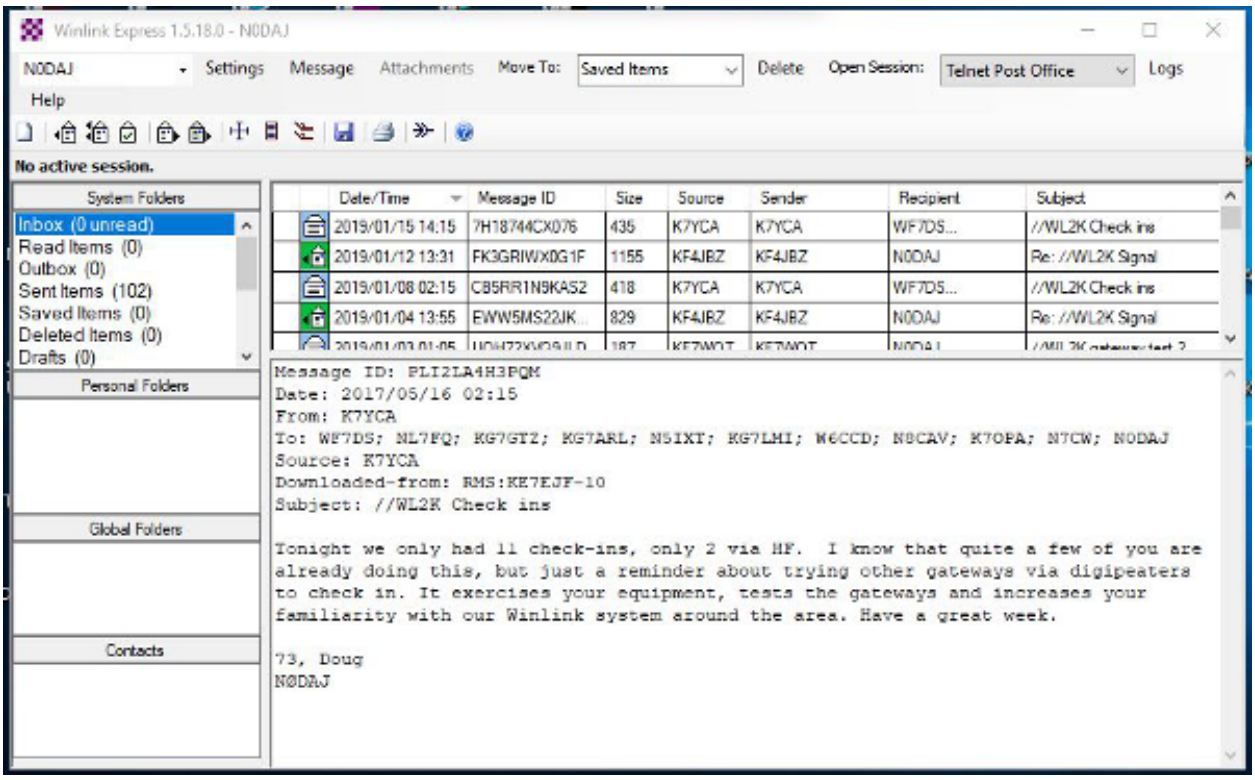
You will need to have detailed knowledge and skills when building your own open source email server, with the advantage of having complete control over everything on the system. There is some administrative overhead for creating and maintaining all user email accounts as well as handling other management tasks on your system. Using these open source software packages, it is possible to build a very robust email server that is capable of running on a small portable computer like a Raspberry Pi.



18.3 Using WinLink to Send Email

Although it is not typically used as a TCP/IP network application, many operators are already familiar with WinLink 2000 for sending message traffic between WinLink computers across amateur radio frequencies. It is possible to configure *Winlink Express* and Telnet Post Office or Telnet P2P for sending email with attachments across a mesh network.

You will need a stable Microsoft Windows computer with plenty of memory to run this system (8GB recommended). The maximum attachment size is currently 5MB per message as compared to the 100KB limitation on HF and Packet RMS stations. Refer to the information below for details about specific network settings and procedures for configuring Winlink over AREDN®. Additional information compiled by Orv Beach W6BI can be found in the document linked here.



18.4 Example Email Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	Features	Network Load	Platform	Effort
Citadel	groupware, webmail	small	lin/mac/rpi	easy
Open Email	client-server	small	lin/mac/rpi	expert
WinLink	email, attachments	small	win (proprietary)	medium

CHAPTER

NINETEEN

FILE SHARING PROGRAMS

File sharing is a method of providing network users with access to digital content. One way to accomplish this is to *push* a copy of a file to users' computers, using either an email attachment or a file transfer program. Another approach is to create a central repository and allow users to *pull* files from this file share. Unless there is a special reason for pushing content, it is usually preferable to let users pull content as needed.

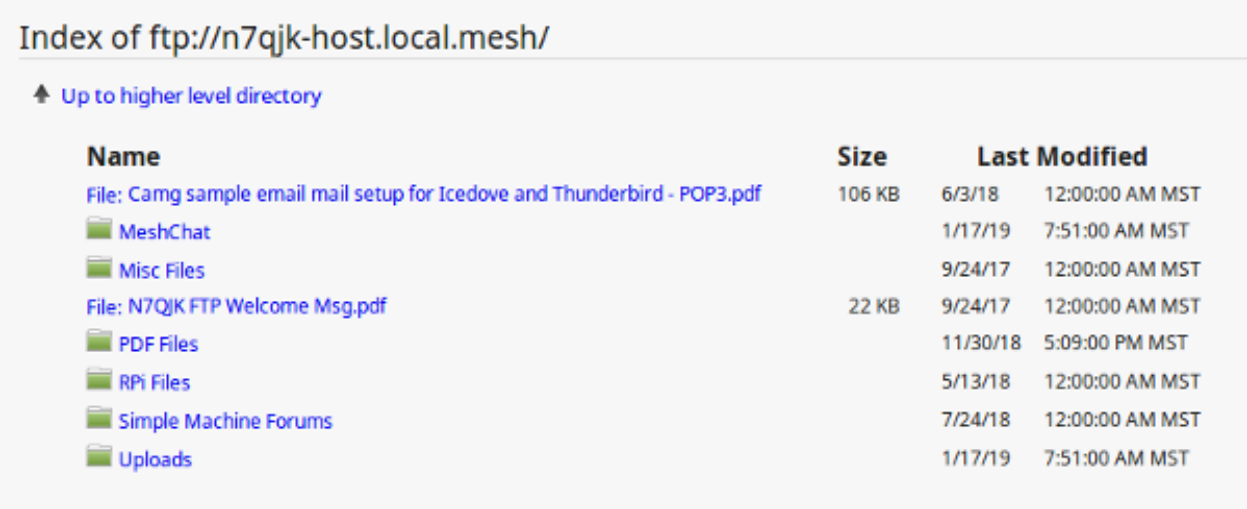
File transfer protocols themselves have minimal impact to network performance, but downloading a very large file across a mesh network could have a major performance impact. Transferring text files, and especially compressed text, should have minimal impact to the network, but a network could experience performance degradation while transferring files with lots of embedded formatting directives or images. High resolution audio files, image captures, or video recordings will also tax network resources when they are moving between nodes.

The following list is not comprehensive or complete but represents a sample of the types of programs that might be available to use for file sharing on your mesh network. Only programs with open source licenses were included in this list, although commercial software can also be used.

19.1 FTP Services

File Transfer Protocol (FTP) servers can be configured as file repositories from which users can copy digital content using FTP client programs. Some of the more common FTP server packages include FileZilla Server, ProFTPD, Pure-FTPd, and vsftpd (which is the default FTP server in many Linux distributions).

All of the most common web browsers allow content to be downloaded using FTP as shown below, although they may not support all protocol extensions. However, there are many FTP client programs with complete FTP support. FTP is a tried-and-true method for retrieving files from a central repository.

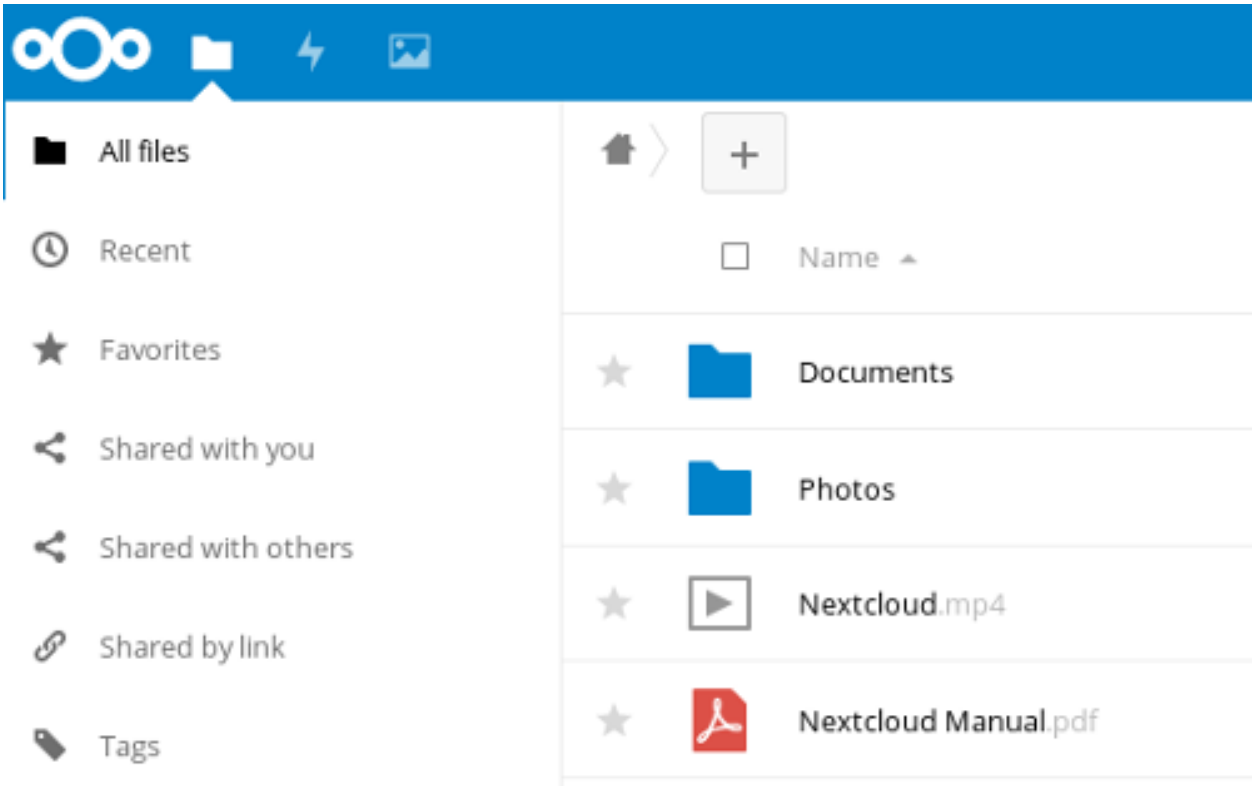


19.2 Web Services

File sharing can be accomplished by hosting downloadable files on a web server. These files can be downloaded from within web browsers using Hypertext Transfer Protocol (HTTP) as well as other built-in file transfer protocols. Simply place files to be shared into the website directory structure and provide links to them on web pages.

There are also many web service packages that provide a robust file sharing interface similar to online cloud storage solutions. One example is NextCloud, an open source file hosting suite with features similar to many of the Internet-based cloud storage services.

Users login to NextCloud to see available content, and file sharing permissions can be set on a user or group basis. Files and folders can be uploaded, downloaded, moved, renamed, deleted, and previewed (depending on file type). Simple file version control is provided through auto-backup, and the *Details* sidebar lists past versions available for rollback. These and other similar software packages can provide a full-featured file sharing service when hosted on a web server.

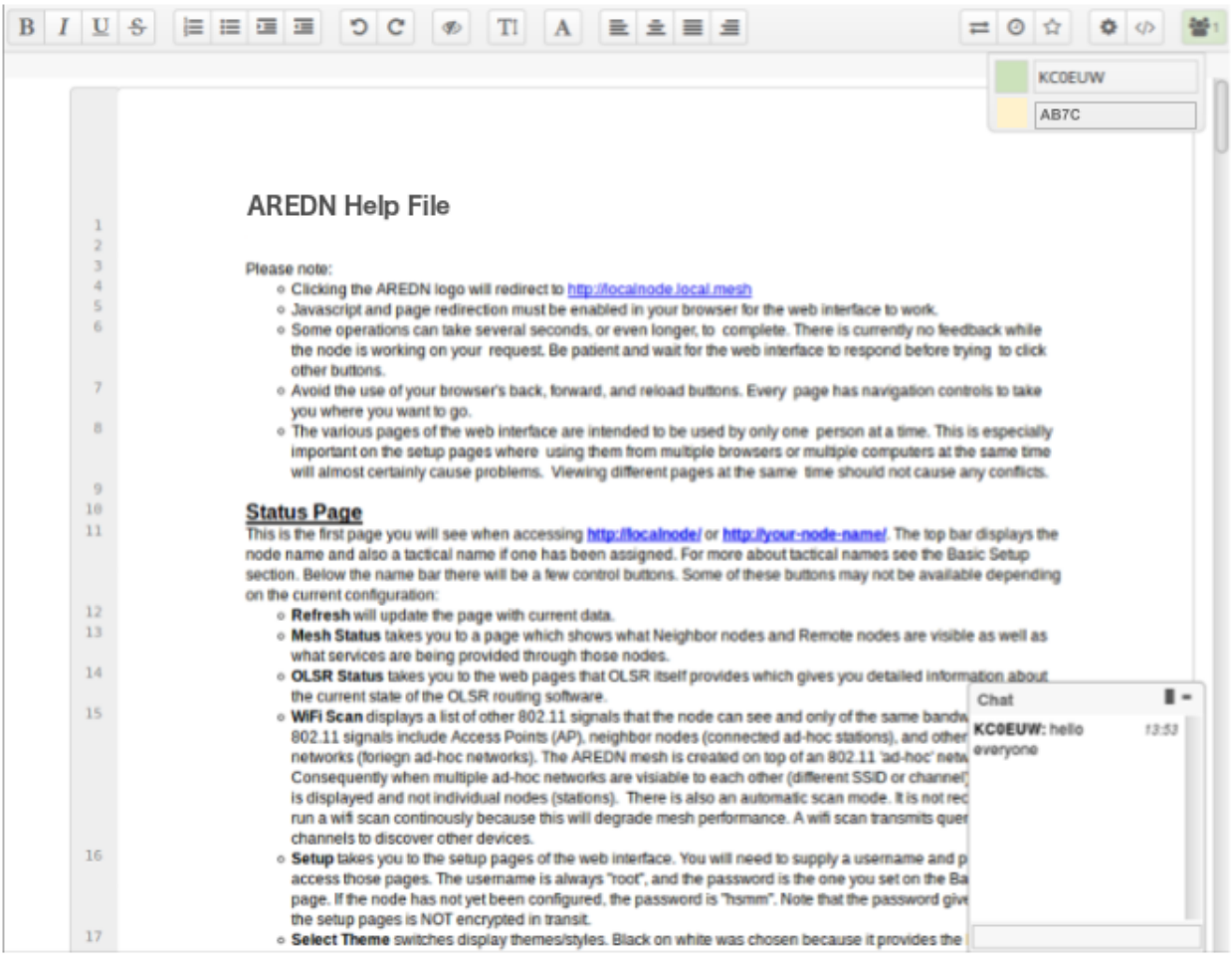


19.3 Collaborative Computing

Collaborative computing enables people to collaborate on documents in real time. Multiple users dispersed across a wide geographic area can be working simultaneously to create or modify a set of documents that are available to others over the network. With this type of collaborative model, documents no longer need be viewed as static but can become truly living projects.

One example package that facilitates collaborative document creation is Etherpad Lite. Users access the Etherpad server through a web browser, so no client software is required on the users' computers. Anyone who connects to the service can create a new document or contribute to an existing document. Active users are displayed and have the ability to chat with each other in the messaging area. Changes to a document are periodically auto-saved, but users can force a checkpoint to capture the current state of a document. The "time slider" control allows users to view document revisions at any point in time throughout its history. Documents can also be downloaded in several formats (text, HTML, Open Document, Microsoft Word, or PDF).

Collaborative document sharing could be very helpful for a number of EmComm use cases, such as maintaining an accurate picture of deployed resources at various locations during an incident or event. Document version tracking makes it possible to scroll back and forth in history to see the status of deployed resources at any given time, as well as to capture information and save it for wider distribution.


KC0EUW
AB7C
AREDN Help File
Please note:
Clicking the AREDN logo will redirect to http://localnode.local.mesh
Javascript and page redirection must be enabled in your browser for the web interface to work.
Some operations can take several seconds, or even longer, to complete. There is currently no feedback while the node is working on your request. Be patient and wait for the web interface to respond before trying to click other buttons.
Avoid the use of your browser's back, forward, and reload buttons. Every page has navigation controls to take you where you want to go.
The various pages of the web interface are intended to be used by only one person at a time. This is especially important on the setup pages where using them from multiple browsers or multiple computers at the same time will almost certainly cause problems. Viewing different pages at the same time should not cause any conflicts.
Status Page
This is the first page you will see when accessing http://localnode/ or http://your-node-name/. The top bar displays the node name and also a tactical name if one has been assigned. For more about tactical names see the Basic Setup section. Below the name bar there will be a few control buttons. Some of these buttons may not be available depending on the current configuration:
Refresh will update the page with current data.
Mesh Status takes you to a page which shows what Neighbor nodes and Remote nodes are visible as well as what services are being provided through those nodes.
OLSR Status takes you to the web pages that OLSR itself provides which gives you detailed information about the current state of the OLSR routing software.
WiFi Scan displays a list of other 802.11 signals that the node can see and only of the same bandw
802.11 signals include Access Points (AP), neighbor nodes (connected ad-hoc stations), and other
networks (foriegn ad-hoc networks). The AREDN mesh is created on top of an 802.11 'ad-hoc' netw
Consequently when multiple ad-hoc networks are visiable to each other (different SSID or channel)
is displayed and not individual nodes (stations). There is also an automatic scan mode. It is not rec
run a wifi scan continously because this will degrade mesh performance. A wifi scan transmits quer
channels to discover other devices.
Setup takes you to the setup pages of the web interface. You will need to supply a username and p
access those pages. The username is always "root", and the password is the one you set on the Ba
page. If the node has not yet been configured, the password is "hsmm". Note that the password give
the setup pages is NOT encrypted in transit.
Select Theme switches display themes/styles. Black on white was chosen because it provides the
Chat
KC0EUW: hello everyone
13:53

CHAPTER

TWENTY

VOIP AUDIO/VIDEO CONFERENCING

The programs described in the previous sections can facilitate the sharing of detailed information across your mesh network. Some of them attempt to emulate a conversation, but nothing can replace an actual interactive discussion. Today people are accustomed to voice conversations, and since much of a message is communicated by non-verbal queues, having an audio-visual conversation can be even more effective. However, these communication advantages come at a cost. Multimedia programs will typically have a much greater impact on network performance than the programs mentioned previously.

The software described in this section can help you to provision services that enable both voice and video conferencing on your network. The phrase Voice over IP (VoIP) encompasses a collection of technologies capable of encoding and delivering realtime multimedia content across a digital network. When you have an established need for this type of communication, and if your mesh network is capable of supporting it, there are many reliable options for implementing VoIP and video conferencing.

The following list is not comprehensive or complete but represents a sample of the types of software that may be available for services on your mesh network. With one exception, programs having open source licenses were included in this list, although software with proprietary licenses can also be used. Dozens of VoIP programs have been available over the years, but the list of current open source projects in active development has dwindled over the past decade. Refer to this link for a comparison of VoIP client and server software.

20.1 VoIP Server

Asterisk Server

Asterisk is one of the original *software* Private Branch eXchange (PBX) servers. It was first designed to run on Linux computers, but it is now available for MacOS and OpenWRT routers. It has been used to build large-scale telephony systems so it has many of the features of commercial and proprietary PBX systems, including voice mail, conference calling, interactive voice response (IVR) menus, and automatic call distribution.

Dozens of full-length books have been written about Asterisk, so it is widely documented.

It also serves as the underlying communication engine for several other software PBX packages. Asterisk is extremely robust tried-and-true IP-PBX software, but you will need specific knowledge and skills to implement it.

FreePBX Server

FreePBX is a web-based graphical user interface (GUI) for managing Asterisk. However, it is most commonly deployed as part of the integrated FreePBX Distro, which installs a complete Linux operating system with Asterisk, FreePBX, and software dependencies included.

All of the extensive features of Asterisk are available along with the benefit of having the FreePBX web interface to facilitate Asterisk management, making it much easier for users who are not telephony experts. Many mesh network operators who deploy VoIP have taken advantage of the *FreePBX Distro* when implementing their PBX services.

20.2 VoIP Endpoints

Once you have a VoIP PBX provisioned on your mesh network, you will need VoIP endpoints which can communicate through the server. Specialized VoIP phone hardware is available from several manufacturers which can provide communication endpoints on your network. It is also possible to use legacy analog phone hardware connected to the network using Analog Telephone Adapters (ATA). In addition to these options, there are pure software phones (softphones) that are supported on a variety of devices, such as the Linphone program described below.

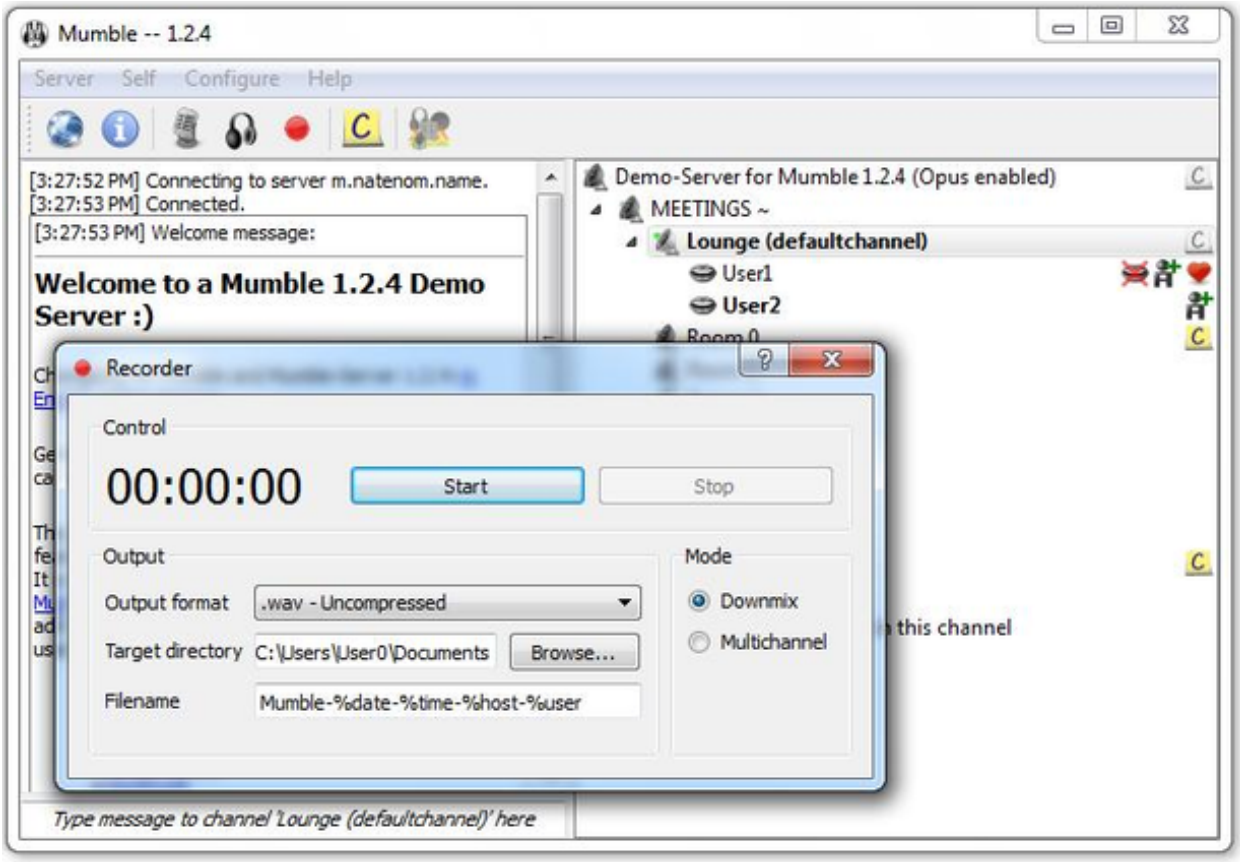
Linphone Softphone

Linphone is a software phone that is supported on Windows, Linux, MacOS, Raspberry Pi, iPhone, and Android. It can be used to place voice and video direct calls as well as calls through a VoIP PBX like those mentioned above. Users can transfer calls to other numbers, send chat messages, share pictures or files, and merge calls into a group conference. The softphone has the ability to manage contact lists, and call history is available for future reference.

Mumble

Mumble is a VoIP package that is available on Linux, MacOS, and Windows systems which support the Qt platform. Mobile apps are also available, such as *Mumblefy* for iPhone and *Plumble* for Android.

Hosting Mumble locally requires downloading the *Murmur* server, which is included as an option in the Mumble installer. The primary users of Mumble are Internet video gamers who want to communicate with each other during game play. However, it can also be used as a non-gaming voice communication service which does not require that an IP-PBX server exist on the network.

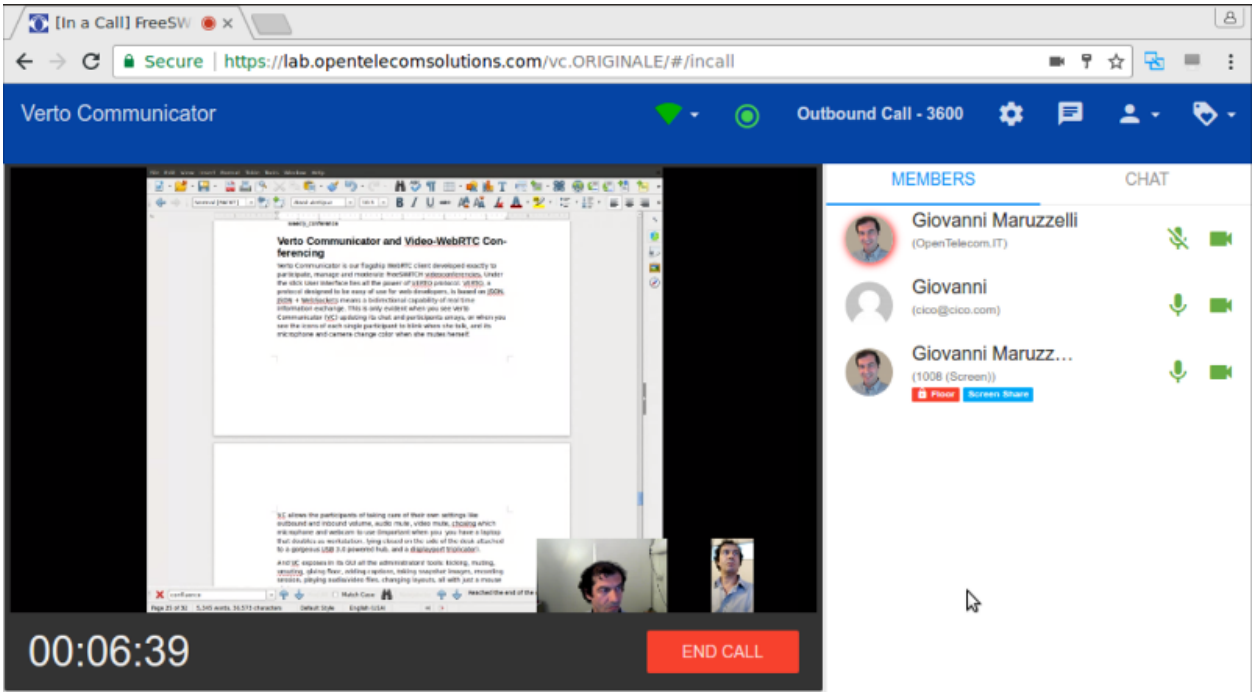


20.3 Video Conferencing Software

FreeSWITCH Server

FreeSWITCH is a recent communication platform that can be used to build voice PBX systems with voice response menus, video conferencing with chat messaging and screen sharing capabilities, and full WebRTC support. Its modular design makes it possible to install only what is required to meet your communication needs. Currently the FreeSWITCH package can be installed on Linux and Windows servers, and it can be compiled on MacOS computers if required.

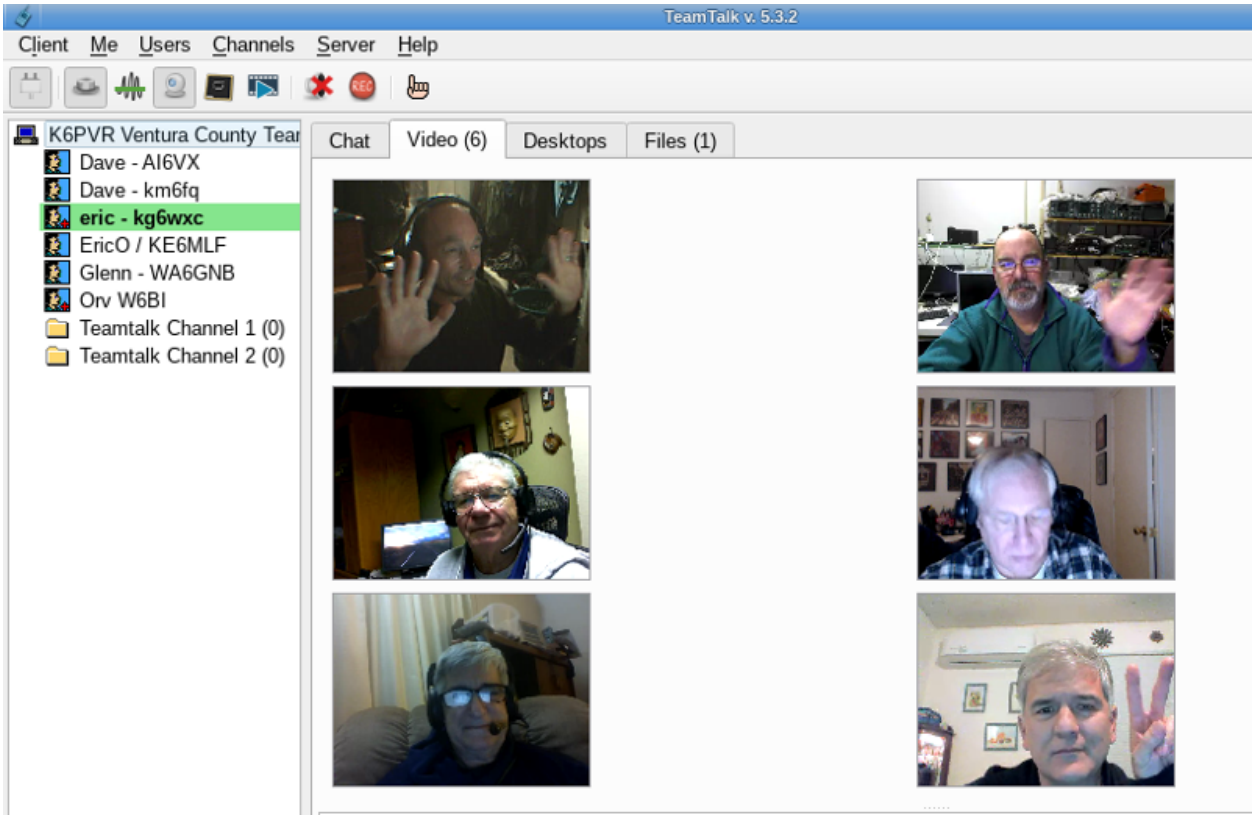
FreeSWITCH provides robust voice and video communication, voicemail, interactive voice response (IVR) menus, user directories, call accounting, screen sharing, chat messaging, call recording, hold music, and many other features that can be implemented as required. It is an extremely flexible communication platform, but you will need specific knowledge and skills in order to install, configure, and manage it as a service.



TeamTalk

TeamTalk is an audio-visual conferencing system which enables people to communicate and share information across the network. It is often classified as *freeware*, but the TeamTalk server is proprietary and its source code is not publicly available. During a conference users talk through their computer microphone, see others via their webcams, create instant messages, share files, and show desktop applications. The TeamTalk software package bundles the client and server programs, so any computer may play the role of client or server.

Voice and video conversations happen in channels or rooms, and a single server can host multiple rooms. While participating in a channel, users can write text messages in the *Chat* tab, view AV webcam streams in the *Video* tab, see shared applications in the *Desktops* tab, and download files from the *Files* tab. The server owner can specify a wide range of access permissions for each available room. TeamTalk is currently supported on Windows, Linux, MacOS, and Raspberry Pi computers.



20.4 Example VoIP Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	Features	Network Load	Platform	Effort
Asterisk	extensive	medium	lin/mac/rpi	expert
FreePBX	web management	medium	lin/mac/rpi	medium
Linphone	client softphone	small	win/lin/mac/mobile	easy
Mumble	voice + chat	medium	win/lin/mac	medium
FreeSWITCH	PBX + video	medium-large	win/lin/mac/rpi	expert
TeamTalk	video conferencing	large	win/lin/mac/rpi	easy

CHAPTER

TWENTYONE

VIDEO STREAMING AND SURVEILLANCE

The previous section described how audio and video traffic can be transmitted across an AREDN® network to facilitate communication. Since these multimedia streams are supported on mesh networks, you can also use them for many other tasks. One example, video surveillance, is often helpful during an emergency or event and AREDN® networks can be used to deliver this type of traffic to Emergency Operations Centers. Keep in mind that multimedia traffic incurs a much greater cost in terms of network performance and computing resources, so be sure your mesh network is designed with the appropriate bandwidth to handle this traffic.

The photo below shows a Mobile Command Center (MCC) deployed to support a large event in San Juan Capistrano, California. An estimated 35,000 people attend this annual gathering, and the local RACES (Radio Amateur Civil Emergency Service) team provides realtime video coverage of the parade route for the sheriff's department and emergency response agencies.

More than a dozen high definition IP cameras were collocated at portable AREDN® node sites across the area, and the individual video streams were consolidated on several large displays in the MCC. Orange County Sheriff's Administrator Sgt. Joseph Cope commented, "This mesh camera system provided by RACES members was a valuable tool for our command staff. The parade was the safest in years. As we were taking the calls, we could see the activity occurring in realtime. Incredibly, there was only one arrest for fighting, which just happened to take place in the camera's view."

21.1 IP Video Cameras

IP video cameras may have a fixed direction and focus, or they may be remote controlled PTZ (Pan,

Tilt, Zoom) models. The cost and features for video cameras vary widely. On the low end is a very inexpensive Raspberry Pi Zero computer having an integrated camera, shown here next to the Ubiquiti Bullet radio. On the high end are the ruggedized commercial PTZ (Pan, Tilt, Zoom) cameras which can cost hundreds of dollars, shown here with the bubble dome and infrared LEDs.

Many IP cameras stream video using Real Time Streaming Protocol (RTSP) in which missing packets are simply skipped during video display. It can be challenging to determine the URL of an RTSP stream, but there is a handy utility at ispyconnect, as well as packet capture utilities such as Wireshark, which may help. Frequently a camera supports multiple RTSP URLs each with a different resolution, so you can advertise any of them as a service on an AREDN® node as required. Recently more cameras support ONVIF (Open Network Video Interface Forum), which is a set of protocols and standards that includes RTSP. It supports camera discovery and PTZ camera control.

A 1920x1080 resolution video stream at 60 frames/second can consume up to eight megabits/second of network bandwidth. Few AREDN® networks can consistently support that load, but lower frame rates reduce the required bandwidth proportionally. Typically 720p at 10 frames per second is more than adequate for video surveillance.

IP cameras with an Ethernet port are preferred in order to simplify network connectivity and ensure adequate data transfer speeds. Configure the camera to obtain a mesh IP address from the node, and reserve the address for that camera in the node's DHCP settings so you have a consistent way to connect to it. A camera with PoE support is also very useful as this simplifies site cabling.

Some cameras are easier than others to configure and deploy, so be sure to research them carefully before investing in expensive camera hardware.

21.2 Video Display Software

The software described in this section can help you to provision video surveillance services on your network. The following list is not comprehensive or complete but represents a sample of the types of software that may be available for services on your network. Primarily programs with open source licenses were included in this list, although software with proprietary licenses can also be used successfully.

21.2.1 iSpy

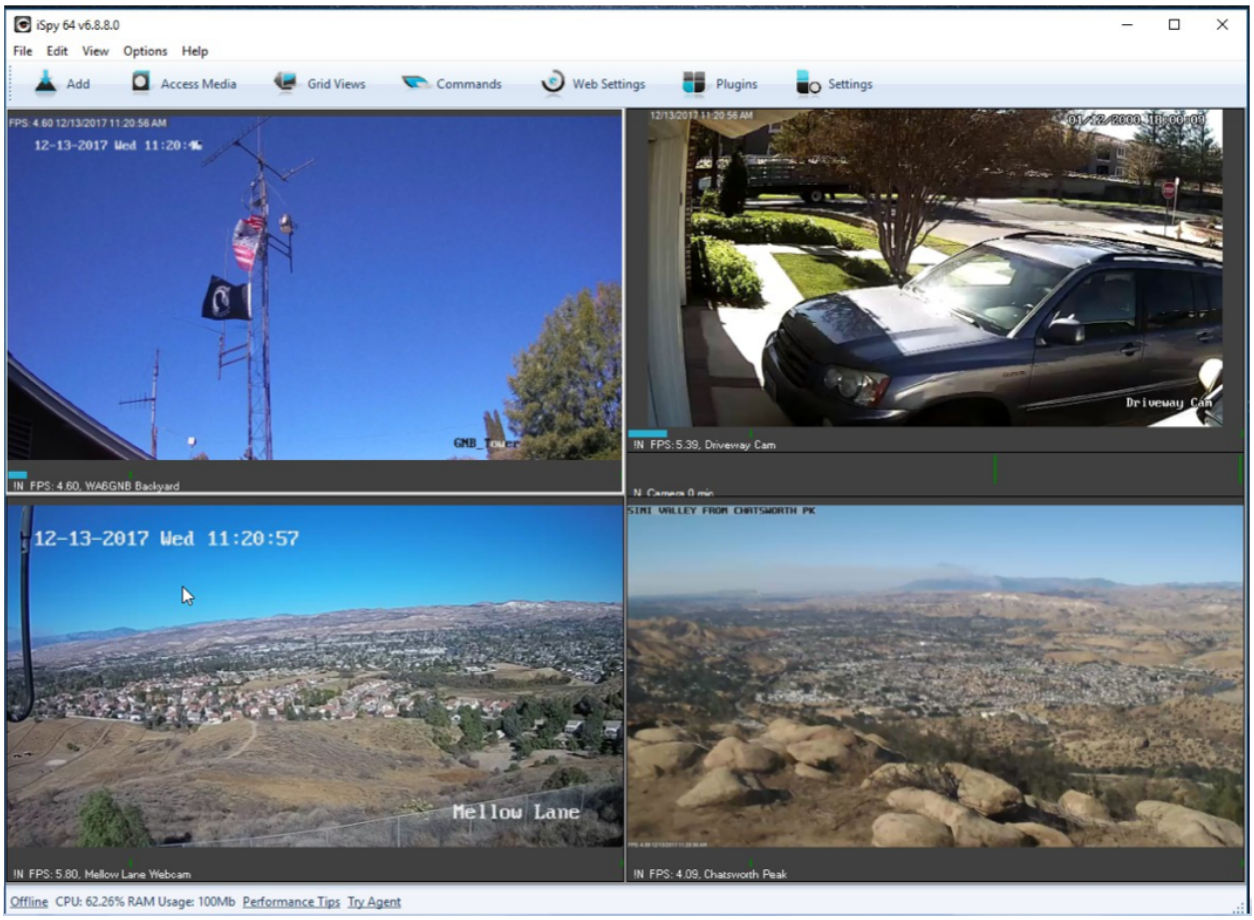
iSpy is a popular video management package for Microsoft Windows computers. It is certified on Windows 7 and above but may work on other systems that support the .NetV4 Framework. iSpy runs as a Windows program with a local user interface (UI) accessible on the computer on which it was installed. Additional services may be available after paying a subscription fee. Parts of the program are licensed under LGPLv3, while other portions are proprietary.

The Windows program provides a "surface" or workspace where you add and configure multiple cameras or microphones. You can then monitor and interact with them to display live video or listen

to live audio from network devices. Multimedia streams can be recorded locally for future use, and PTZ cameras can be manipulated with controls in the UI. Motion detection can also be configured, which provides a method for automatically recording multimedia snippets when specific events occur.

iSpy can connect to IP cameras using MJPEG or JPEG sources. It also supports camera connections using MP4, ASF, or RTSP, which it accomplishes through a VLC plugin after Videolan software is installed. VLC requires usernames and passwords directly in the URL, so you must enter them in clear text as in this example: `http://admin:password@192.168.1.4/video.asf`.

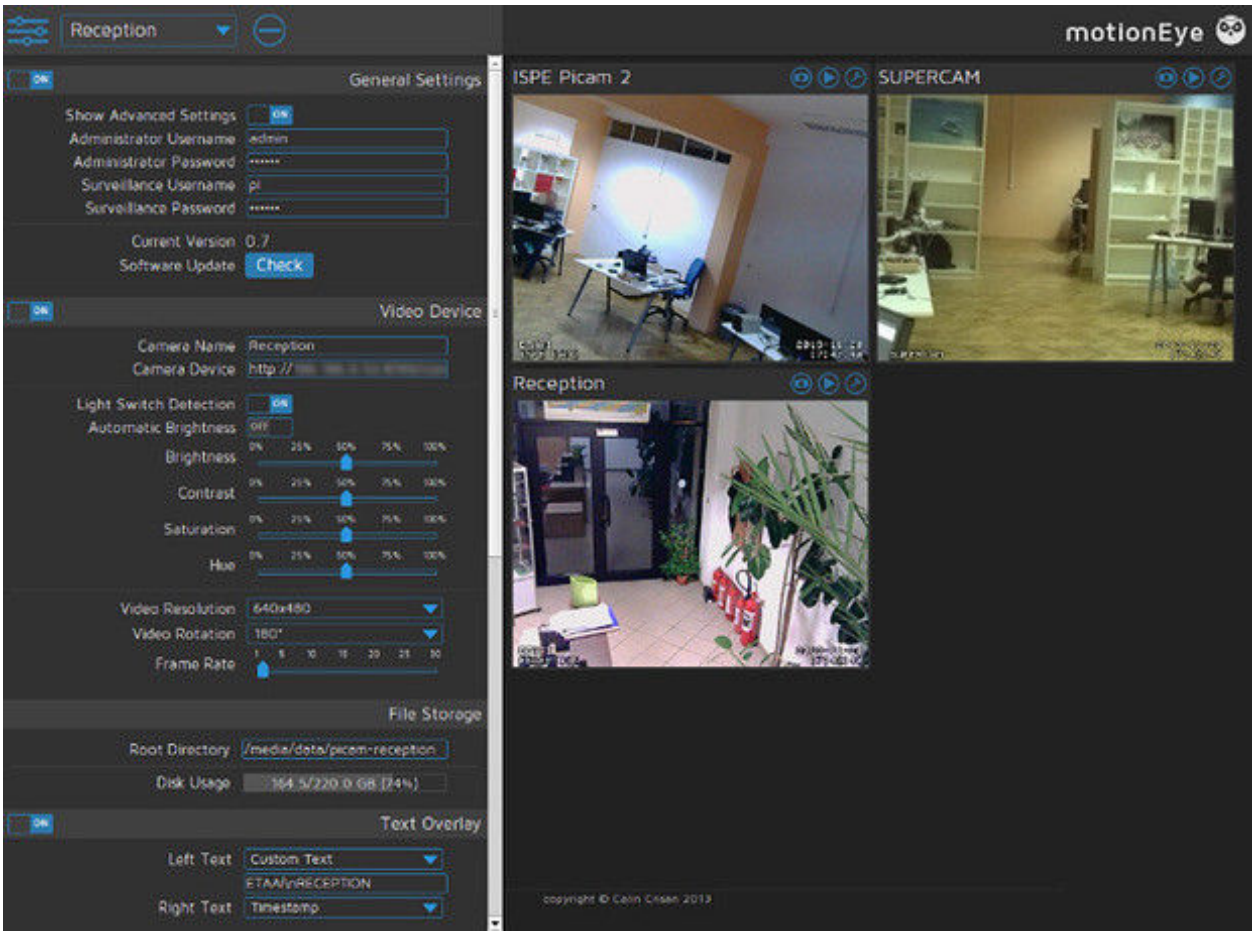
In the lower right video stream on the iSpy display below you can see the smoke plume from the 2017 Thomas Fire in California, which was recorded by a camera on the local AREDN® network. For additional information about iSpy, visit this link: iSpy.



21.2.2 MotionEye

MotionEye is a lightweight video display program which runs on Linux and Raspberry Pi computers. It can connect to a variety of USB or IP cameras, and it has the ability to display video streams in a grid format accessible by any web browser on the mesh network. Authentication as a regular user or an administrator will display different menu options: view options for regular users or full administrative control for admin users.

The backend Motion engine is built to provide robust motion detection and event triggering. It also enables custom scripts to extend its features, for example to print the system temperature and update it every ten seconds on the display. Many AREDN® operators implement MotionEye on low-power portable Raspberry Pi computers, and the MotionEyeOS distro installs the operating system with all dependencies on this platform. For additional information about MotionEye, visit this link: MotionEye

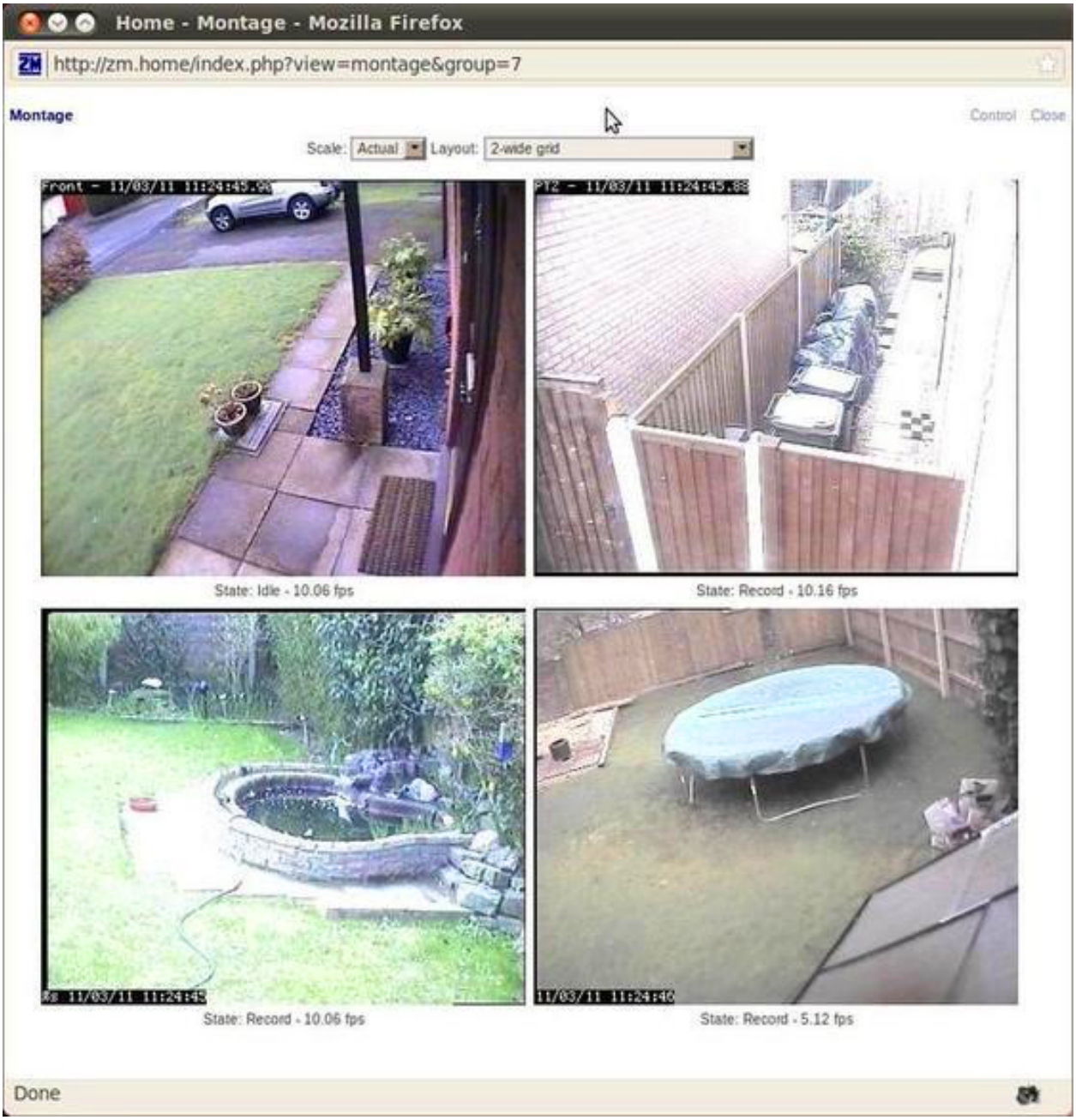


21.2.3 ZoneMinder

ZoneMinder is a full-featured video package which runs on Linux computers. Its display is accessible across the mesh network by web browser. IP cameras are supported which use MJPEG streams or an interface to JPEG images. Camera connections can be configured for monitoring, recording, motion detection, or a combination of these.

The ZoneMinder name comes from the fact that it allows administrators to define "zones" or regions of an image, each with different motion detection sensitivity levels. During motion detection, each frame is compared with previous frames and checked for differences. If the amount of change is greater than a specified percentage, an event will be triggered which can capture recordings, send email alerts, or execute external programs. ZoneMinder has extensive features for filtering and comparing video images, which can be useful for monitoring a high traffic area with a single point of interest such as an entry door next to a busy walkway.

This robust feature set comes at the cost of some administrative complexity, making ZoneMinder a good candidate for operators with skills and experience in Linux and video systems. Its open design and the ability to execute external programs makes ZoneMinder very flexible for integration with other systems. For additional information about ZoneMinder, visit this link: ZoneMinder.



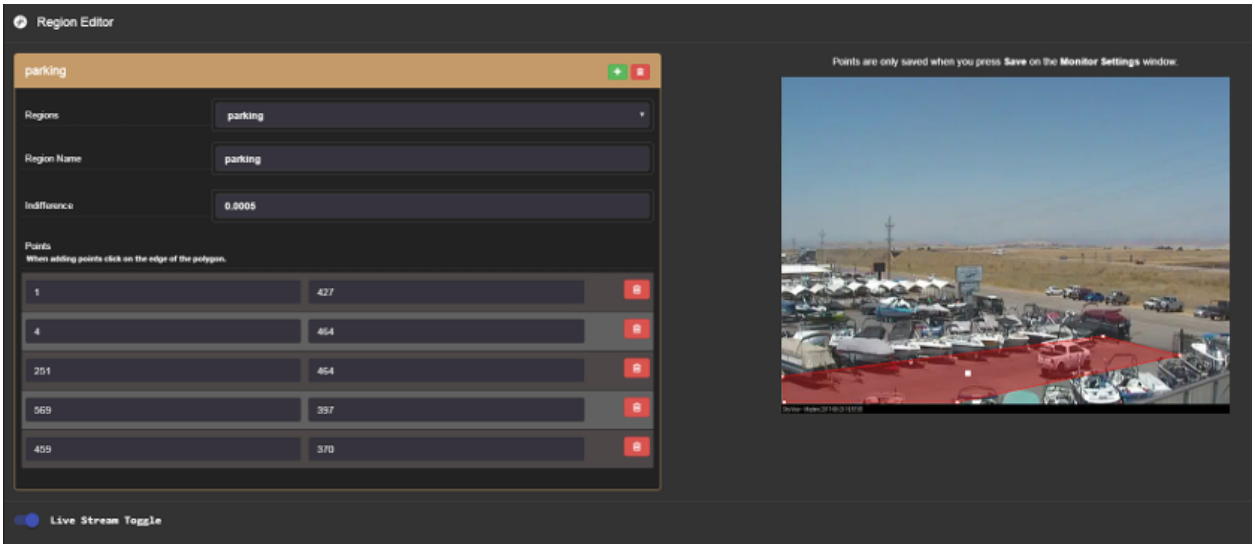
21.2.4 Shinobi

Shinobi is a fairly recent video project which implements current methods of streaming for the web. It supports legacy MJPEG/JPEG, FLV, and RTSP streams as well as the newer HLS and Websocket methods. The web browser interface (UI) is clean and responsive, which renders well on tablets and mobile devices. It is designed for ease of navigation, with dropdown and pop-up menus for snapshots, video recording, event lists, and configuration options.

ONVIF (Open Network Video Interface Forum) compliance allows Shinobi to provide PTZ camera

controls. Motion detection is accomplished through plugins, with regions configured in the web UI, so if you do not require motion detection you can conserve resources by not adding it to your system. There are three user levels which provide delegation of authority: Superuser, Admin, and Sub-account. Superusers control system settings and create Admin accounts, which control camera settings and manage Sub-accounts and Groups. Sub-accounts have limited privileges and camera profiles can be shared by Group members.

Shinobi tends to conserve computing resources fairly well, so more cameras or higher resolution streams could be supported on a server. The image below shows how motion detection regions are defined, in this case to monitor traffic along an access road to a parking area. For additional information about Shinobi, visit this link: Shinobi.



21.3 Example Video Service Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	License	System Load	Platform	Effort
iSpy	freemium	large	windows	easy
MotionEye	open source	medium	lin/rpi	easy
ZoneMinder	open source	large	linux	expert
Shinobi	free for *NC* use	medium	lin/mac	medium

NC ~ non-commercial

CHAPTER

TWENTYTWO

NETWORKING TOOLS

There are several service programs that can assist in visualizing or mapping an AREDN® network, as well as for viewing local RF conditions near your node. Some of these programs are discussed below.

22.1 Manage Extra Static Routes

There may be cases when you need to create extra static routes to control the flow of network traffic through your node. You can maintain your extra routes by entering them into the `/etc/aredn_include/static_routes` file. Login to your node at the command line to edit the routes in this file. After saving your changes you should run `/usr/local/bin/node-setup` to apply your changes, then reboot your node. You can view the OpenWRT Static Routes page for additional information about managing static routes.

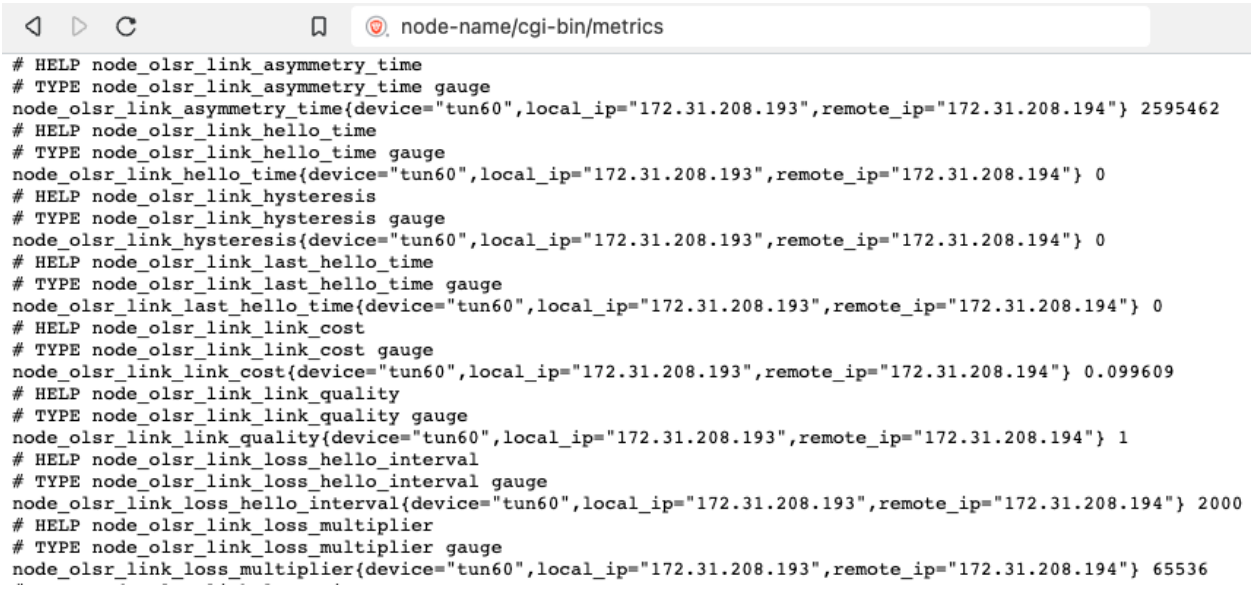
22.2 AREDN® Prometheus Exporter

Prometheus is an open-source monitoring and alerting toolkit which collects and stores metrics as time series data. Prometheus evaluates rule expressions, displays the results, and can trigger alerts when specified conditions are detected. It can collect metrics from AREDN® nodes running a recent firmware version.

Examples of AREDN® metrics include:

- Node details (name, model, firmware, description, Lat/Lon, grid square, band, channel, width, frequency, SSID)
- Memory, storage, CPU, networking, and per-process metrics
- RF metrics (signal, noise, MSC rate, TX/RX packets/rates)
- LQM tracker metrics
- Link info

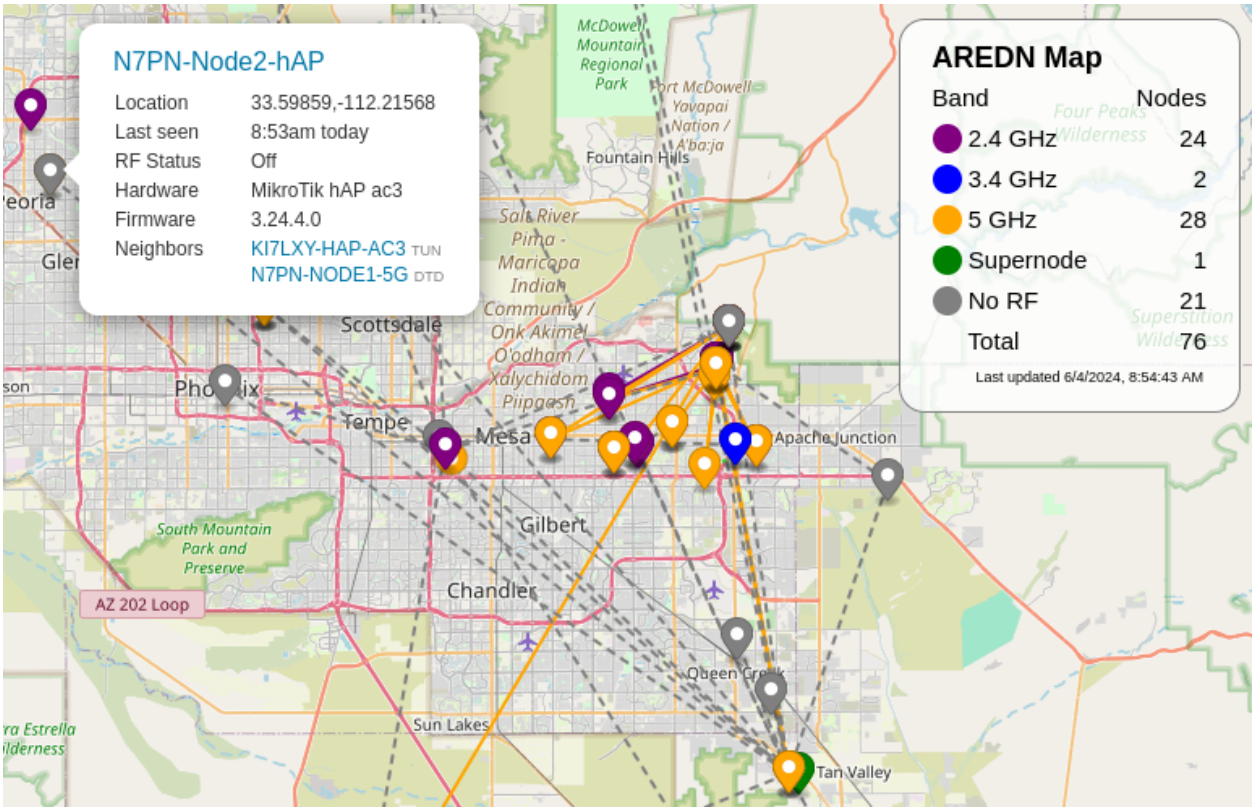
In order for Prometheus to pull metrics from a node it will use the following target URL: `http://<NODE>.local.mesh/cgi-bin/metrics`, and metrics are returned by the node as standard *text/plain* content. Minimal node resources are required to support Prometheus data collection since the node runs no metrics service and uses minimal resources when its URL is queried.



The AREDN® node simply makes these metrics available for Prometheus to pull. For additional information about Prometheus itself, visit their website here. The following image shows Prometheus metrics for an AREDN® node being displayed by the Grafana visualization application.

22.3 KN6PLV Mesh Map

Tim KN6PLV created these programs to discover and visualize your mesh network. They can be installed on one of your LAN-attached computers that is running a web server. This software is available for download here: KN6PLV NewMeshMap. Once you have followed the install instructions and have a working mapper, you will be able to view your mesh network in a web browser (as shown in the example below).


N7PN-Node2-hAP
Location 33.59859,-112.21568
Last seen 8:53am today
RF Status Off
Hardware MikroTik hAP ac3
Firmware 3.24.4.0
Neighbors KI7LXY-HAP-AC3 TUN
N7PN-NODE1-5G DTD
AREDN Map
Band Nodes
2.4 GHz 24
3.4 GHz 2
5 GHz 28
Supernode 1
No RF 21
Total 76
Last updated 6/4/2024, 8:54:43 AM
McDowell Mountain Regional Park
Fort McDowell Yavapai Nation / A'ba:ja
Fountain Hills
Salt River Pima - Maricopa Indian Community / Onk Akimel O'odham / Xalychidom Piipaash
Scottsdale
Tempe
Mesa
Apache Junction
Gilbert
Chandler
South Mountain Park and Preserve
AZ 202 Loop
Queen Creek
Sun Lakes
Tan Valley

CHAPTER

TWENTYTHREE

COMPUTER AIDED DISPATCH

Computer Aided Dispatch provides an automated way for emergency services agencies to keep track of incidents, activities, information, tasks, messages, and the status of deployed resources. Command staff are able to see the big picture, while at the same time maintaining detailed records of plans and actions for future reference. Deployed resources are able to clearly communicate in realtime, while having much better situational awareness of surrounding events.

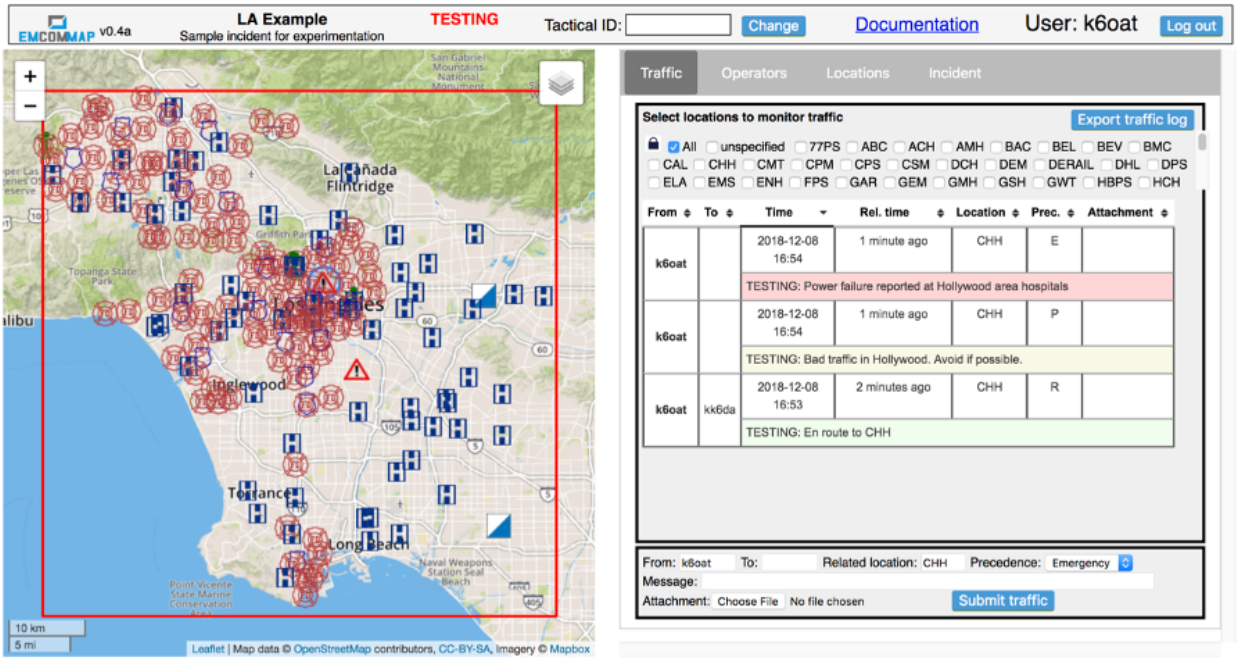
Served agencies have been using Computer Aided Dispatch (CAD) software for quite some time, and it has become their preferred method for managing events and incidents within their jurisdiction. In emergencies when electrical power or mission-critical facilities become unavailable and agencies are forced to operate off-grid, AREDN® operators with portable power for mesh networks and computing resources can bridge the gap by providing CAD (Computer Aided Dispatch) solutions for personnel at key sites.

There is a wide variety of CAD software in use today. Many of the sophisticated commercial packages have integrated automatic vehicle location (AVL) and geographic information systems (GIS) which require large amounts of network bandwidth and dedicated computing resources that might not be accessible during an emergency.

The programs described in this section can help you to provision CAD services for emergency use on your mesh network. The following list is not comprehensive or complete but represents a sample of the types of software that may be available for services on your network. Programs with open source licenses were included in this list, although software with proprietary licenses can also be deployed.

23.1 EmComMap

EmComMap was designed by an Amateur Radio Emergency Service operator for use on AREDN® mesh networks during deployments. It leverages modern technologies for interactive maps and sync-able web browser databases to enable map-based situational awareness and emergency communication across IP networks. Based on this architecture, EmComMap is one of the more mesh-friendly CAD programs with additional features in progress for data distribution.



A specific geographic region is defined within which an incident is in progress, and the location of resources are shown on the map using icons *(Police, Fire Department, Hospital, Government Facility, Incident Command Post, EmComMap Node)*. Each map can be zoomed and panned as required to view location details for all deployed resources. Incident information can be defined and updated on the *Incident* tab, while locations are defined and updated on the *Locations* tab. Message traffic is available to all operators across the network on the *Traffic* tab, and operators update their location and status on the *Operators* tab. Open Street Map tiles can be downloaded to the server for standalone operation.

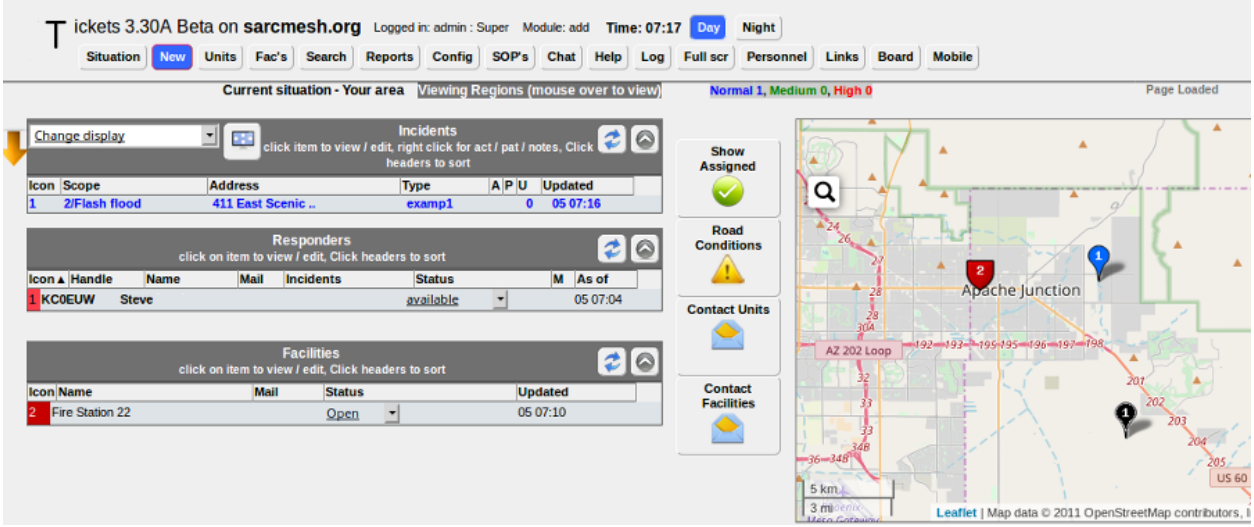
All communications are tracked and can be exported in spreadsheet format for offline use. Message traffic can be filtered to view specific messages for selected locations, and the traffic table can also be sorted for viewing the details based on information in any column. Message severity levels and tactical call signs are supported, and operators are allowed to send messages and report status information on behalf of other users if necessary. EmComMap is a recent program under active development, with continual feature improvements in progress. For additional information about EmComMap, visit this link: EmComMap.

23.2 Open ISES Tickets

The *Open Information Systems for Emergency Services* (ISES) project is a community of software developers, paramedics, EMTs, law enforcement, and fire fighters working to create software and training materials for the emergency service community. They currently offer the *Tickets* CAD system, which has an extensive suite of features that are accessible by web browser from a mesh network server. Any computing platform is capable of running a *Tickets* server if it supports the traditional LAMP, XAMPP, or MAMP packages.

Tickets presents a situation dashboard showing incidents, responders, and facilities along with a GIS map of their locations. Open Street Map tiles can be downloaded for standalone operation. Clicking any of the controls allows operators to drill into item details, and *Tickets* provides database tracking for a large array of information about each item. The dashboard can be fully integrated with several different functions, including email, chat, routing, and tracking (for example, with Automatic Packet Reporting System [APRS]).

A variety of built-in reports are available which can be viewed, printed, and downloaded for distribution. Standard ICS forms are available for online completion and emailing, and custom *Standard Operating Procedure* (SOP) documents can be integrated for viewing through dashboard links in the web browser. For additional information about *Tickets*, visit this link: Open ISES Tickets.



23.3 Example Computer Aided Dispatch Comparison

Platform abbreviations:

win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

Program	License	System Load	Platform	Effort
EmComMap	open source	small	linux	medium
ISES Tickets	open source	small	win/lin/mac/rpi	medium

CHAPTER

TWENTYFOUR

OTHER SERVICES

As mentioned in the *Services Overview*, almost any program that can operate across a peer-to-peer TCP/IP network is a candidate for AREDN® networking. Many useful services have been discussed previously, and this section will list some of the other types of services that you might consider deploying on your mesh network.

24.1 Network Time Services

There are programs or services running on your node and network which would benefit from having accurate network time updates. Network Time Protocol (NTP) is a reliable way for networked devices to update their system clocks. It may be important to have accurate timestamps across the network for services such as Wireguard, MeshChat, email message logging, file timestamps, video surveillance images, and many others.

Most NTP implementations depend on an Internet connection in order to synchronize with upstream

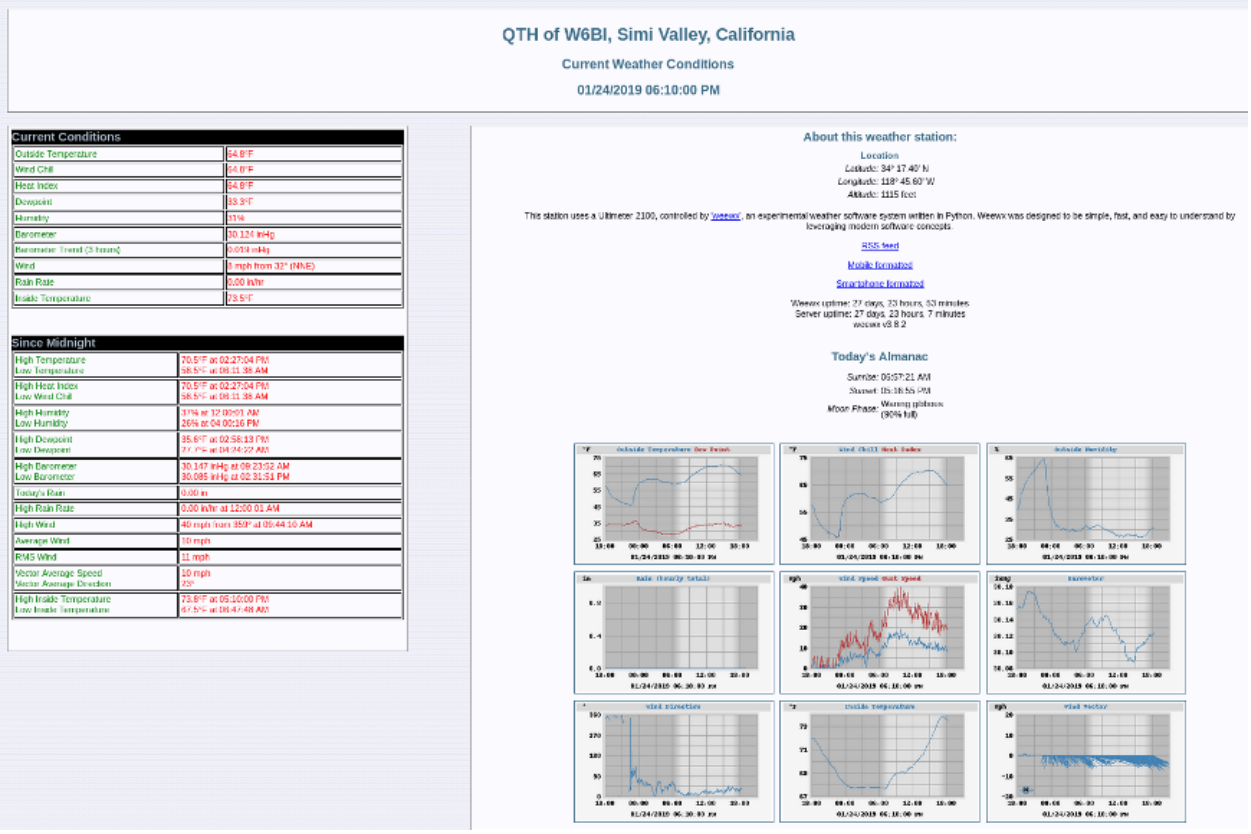
time servers. To synchronize system clocks in an off-grid situation, one or more battery powered devices can be configured as NTP servers which retrieve upstream time from GPS satellites *(stratum 0)*.

Position your portable NTP server so that it maintains a clear view of the sky and gets a fix on as many GPS satellites as possible. In order for NTP to operate properly, each client device must have a reliable connection to the NTP servers on the network. Be sure to locate your NTP servers on reliable high-speed segments of your mesh.

You may choose to purchase an off-the-shelf NTP appliance such as those offered by Centerclick and others. There are also many sources of information for building your own off-grid NTP server (for example, this one using a Raspberry Pi: G4WNC NTP)

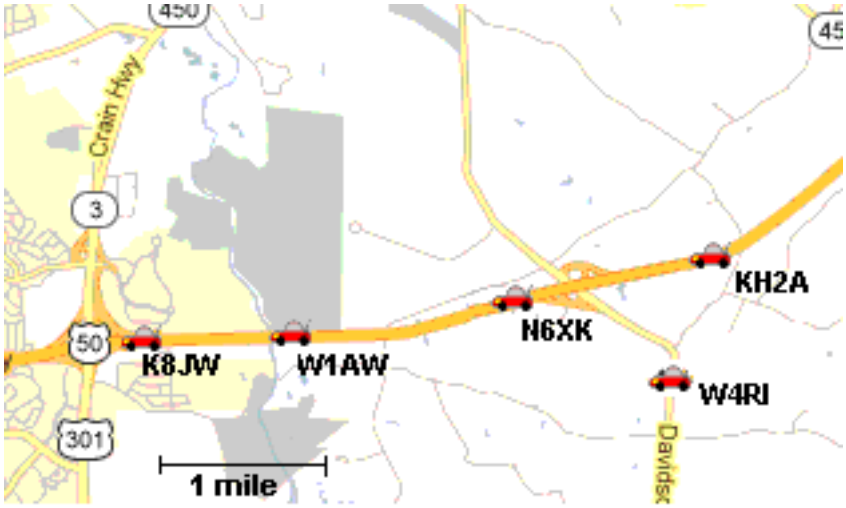
24.2 weeWx Weather Service

Many operators have weather stations, as do quite a few repeater sites. If those weather stations can be put on the mesh network, they can provide a valuable overview of weather conditions across a wide area, for example, showing wind speeds and rainfall totals for each location. The *weeWx* package is available for many different operating systems and weather station models. It supports serial, USB, and Ethernet connections to weather stations. For additional information about weeWx, visit this link: weeWx.



24.3 GPS Tracking Services

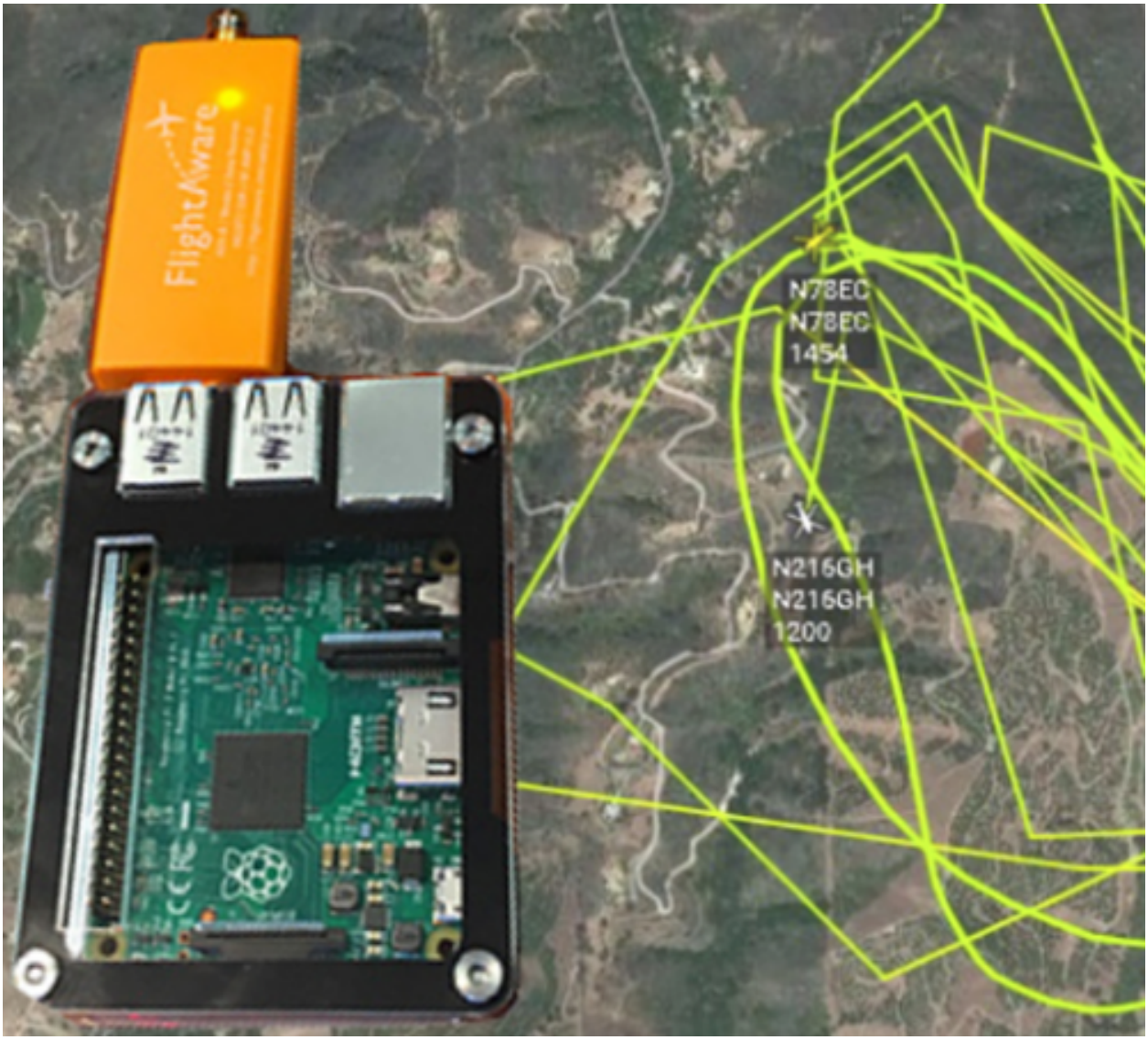
Tracking deployed resources is an important task during any emergency. There are many options for monitoring and displaying the GPS locations of tracked resources, two of which are mentioned here.



Many amateur radios and portable locating beacons transmit Automatic Packet Reporting System

(APRS) information. It is possible to implement an APRS receiver using inexpensive, battery-powered, portable computers and USB Software Defined Radios (SDR). The details are widely available for building these receivers using Raspberry Pi computers with Direwolf and Xastir or YAAC software.

There may be situations when it would also be helpful to track the locations of aircraft during an emergency. Automatic Dependent Surveillance-Broadcast (ADS-B) information is available which can be captured using portable computers with ADS-B receivers. The following image shows the track of two water tankers dropping fire retardant above Santa Barbara, California, during the 2017 Thomas Fire. This information was displayed across an AREDN® network using an ADS-B Ground station which was running as a mesh network service.



Depending on the requirements of your specific situation, almost any program that can operate across a peer-to-peer TCP/IP network could be deployed as a service on your mesh network.

CHAPTER

TWENTYFIVE

BEGINNER'S GUIDE

Contributor: Orv Beach W6BI

25.1 What it's all about

By loading the AREDN® firmware in a outdoor wireless access point, you can join a ham radio network. It's like the Internet but runs on ham radio frequencies, mostly in the 5.8 GHz band. The network is growing steadily, but the number of accessible nodes in your area can range from plentiful to none within hundreds of miles. By joining this network you can find and use all sorts of applications (known as "services"). Anything running on a server, like weather stations, web sites showing site conditions, and email servers can be provided as a service. There are also services that don't rely on a browser: video streams, chat servers, and VOIP PBXes. The network can also be used to connect Winlink stations, Dstar and DMR repeaters, and Allstar devices. Pretty much any kind of service you can put on the Internet you can put on the AREDN® network, subject to the restrictions of the ham radio regulations (FCC "Part 97").

25.2 Getting Started

The network is growing steadily, but the number of accessible nodes in your area can range from plentiful to none within hundreds of miles.

25.3 RF access to the network

How do you find out where the nearest network node is?

There are several ways:

- Check the worldwide map at worldmap.arednmesh.org.

- Create an account on arednmesh.org and see if there's a Regional Forum for your area. Ask there for local information. If you don't find a regional forum for your area, get it started by asking the webmaster, Randy WU2S, to create one for you. Search for his callsign using the search function, and by finding it you'll be able to drop him a note. Once it's created, create a post mentioning you'd like to get started and asking if there's any activity near you.
- Ask around your local ham club or on a local net.

If you determine there is a local node, how do you find out if you can reach it?

For these devices, line of sight is REALLY line of sight; they don't do trees well at all. There are a number of online LOS calculators, ranging from simple to use to really complex. A simple one is at https://heywhatsthat.com. By entering your location, expected elevation of your node, and naming it, the site will generate a coverage plot for you. You can do this for your QTH and/or any existing local nodes. (See the *Network Design Guide* for other tools.)

25.3.1 Alternative to RF access

If after doing your research you find that you don't have any RF path to the network, don't despair; there is an alternative. The nodes have the capability of 'tunneling' over the Internet to another node. While this isn't a radio connection, it will let you get on the network until such time as the network has grown into your area.

In order to establish a tunnel, you'll need an additional piece of network equipment beyond the node itself. The current preferred device is the `Mikrotik hAP ac3` router which when running AREDN® firmware will provide your node access to the Internet, plus a host of other really useful features when running a ham network in your shack.

25.4 Recommended equipment

The following recommendations are for a home station. Recommendations for network nodes on hilltops are likely to be different and beyond the scope of this introductory article. In order to ensure good performance you need a strong RF link to the network. Like most other ham radio activities, more gain is better than less. And even if you have two nodes within range the node's routing software will always pick the strongest one as your path into the network. Omnidirectional antennas are discouraged and dish antennas greatly preferred.

All of the equipment using dish antennas supported by AREDN® use electronics integrated into the feed point: two transceivers, two modems, and an embedded computer with RAM, ROM, and a network interface. They are all POE-enabled (Power Over Ethernet). This avoids having to run both a network cable and a power cable up a mast to the node. A web interface is used to control and configure the device. Your equipment will need to be on the same band as the node you want to connect to. If there are both 2 GHz and 5 GHz nodes equidistant from you with similar path characteristics, choose 5 GHz. That band is quieter, there's more bandwidth, more channels, and the gear is about the same cost.

Fortunately AREDN® now supports dozens devices (including new ones added in the nightly builds), so there are many options when choosing what to buy. Purchasing legacy 802.11n devices, although many of them are still supported by AREDN, is no longer recommended. Instead, consider 802.11ac devices – being a newer generation of hardware they have faster CPUs, more RAM and Gigabit Ethernet ports, resulting in faster throughput.

25.4.1 Short haul options

- Mikrotik SXTsqG-5acD 16 dBi gain (~10-12 miles)
- Ubiquti Nanostation 5ac 16 dBi gain (~ 10-12 miles)

25.4.2 Long haul options

- Ubiquiti LiteBeam ac gen2 23 dBi gain
- Ubiquiti PowerBeam ac 500 25 dBi gain
- Mikrotik RBLHGG-5acD 24.5 dBi gain
- Mikrotik RBLHGG-5acD-XL 27 dBi gain

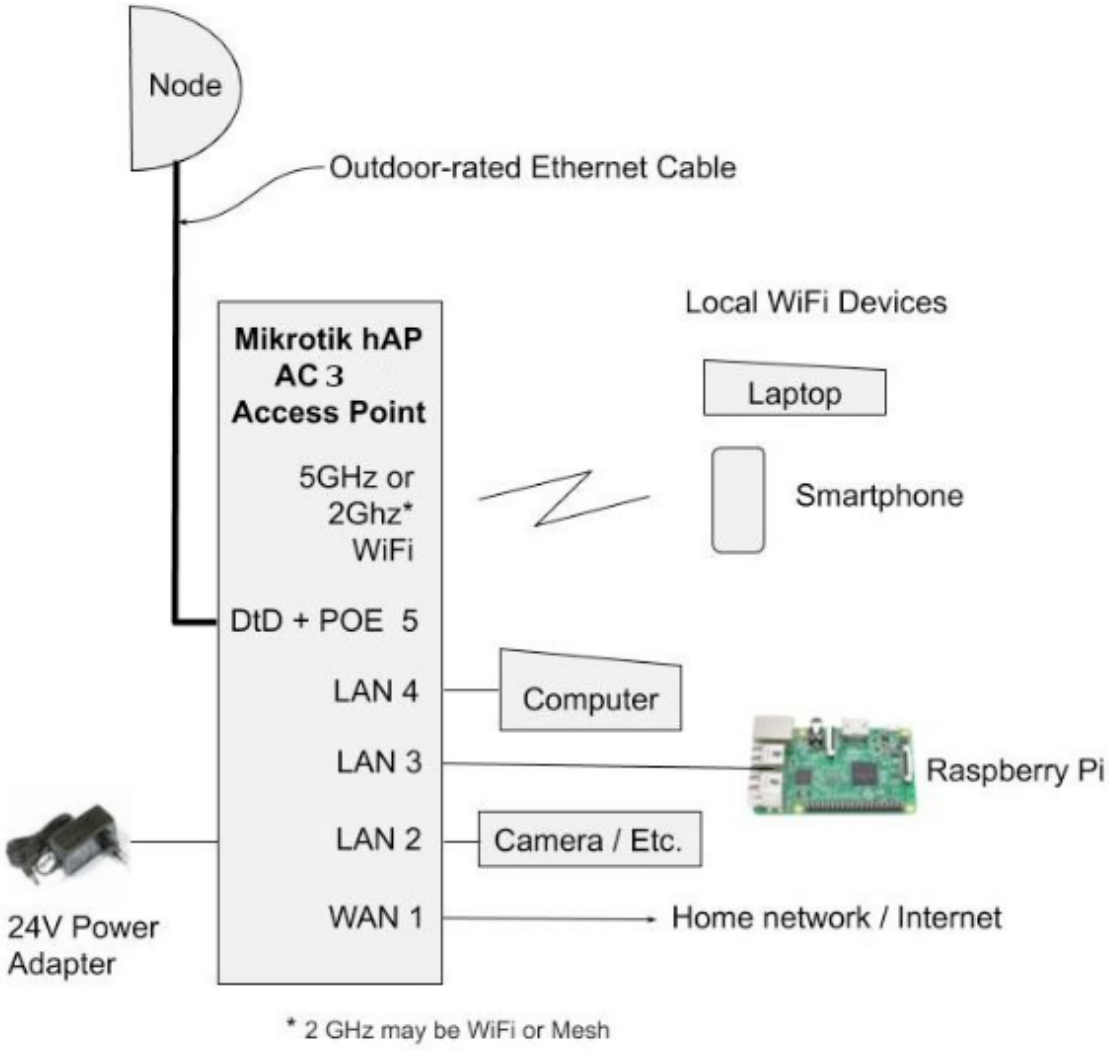
The Mikrotik dishes may be slightly less durable than equivalent Ubiquiti dishes, so if you live where there's severe icing choose more rugged equipment. Mikrotik dishes are lighter, making them more suitable for portable work (e.g., go boxes), but both brands work well. The AREDN® team has flagged *Sunset devices* that are no longer recommended for new purchase in the Supported Platform Matrix:.

25.4.3 In the shack

The **Mikrotik hAP ac3** (RBD53iG-5HacD2HnD) is a five-port router. It provides a seamless method for integrating the ham radio network into your ham shack network. When running AREDN® firmware, its ports can be configured to provide any of the following functions:

- DTD = Device to Device port to connect directly to another AREDN® node.
- WAN = to connect to your home router for Internet access.
- LAN = to connect your shack PC to both the Internet and the mesh network, eliminating the need for two computers in the shack, one on each network. Two spare ports can be for things like cameras, VOIP phones or Raspberry Pi computers.

Over and above those features, the *hAP ac3* has two internal radios (2.4 and 5.8 GHz). Either can be configured for mesh RF or as a wireless access point. Having a wireless Part 15 access point on your shack's ham network is very handy, since you can link your laptop or smartphone to it wirelessly and have access to both the Internet and the mesh network.



25.5 Configuring your node

After you have your equipment in hand, you need to install the AREDN® firmware, configure its settings, and put it up in the air. Installation and configuration of the firmware is covered in the **Installing AREDN® Firmware** and **Firstboot Node Setup** sections of the *Getting Started Guide*.

25.6 Aiming high gain antennas

Note that the higher the gain, the narrower the beamwidth and the trickier it is to aim these dishes accurately. Fortunately, some aiming tools have been added to the AREDN® firmware that help in setting up the dish in the correct direction and elevation. Remember that the vertical beamwidth is as narrow as the horizontal beamwidth. Review the **Tips for Aiming Directional Antennas** document in the **How-To Guides** section for more information. Do not stand in front of the antenna or dish for extended periods of time when it's powered on. NEVER look into the focus of the radio when it's powered on. These small dishes have 80 - 100 watts of ERP at 5.8 Ghz! The Mikrotik Basebox 2 has 30 dBm of power output. When fed to a Mikrotik 30dBi gain dish, that's 1 KW of ERP. Use caution!

25.7 Tools for planning your network

The **Network Modeling** section of the *Network Design Guide* describes several external tools that may be helpful for planning your network. These include calculators for determining and visualizing a radio path, the Fresnel zone, and using computer modeling to estimate network coverage.

25.8 Example node deployments

Here are some typical deployment scenarios for connecting an AREDN® node with PoE power adapters and computers.

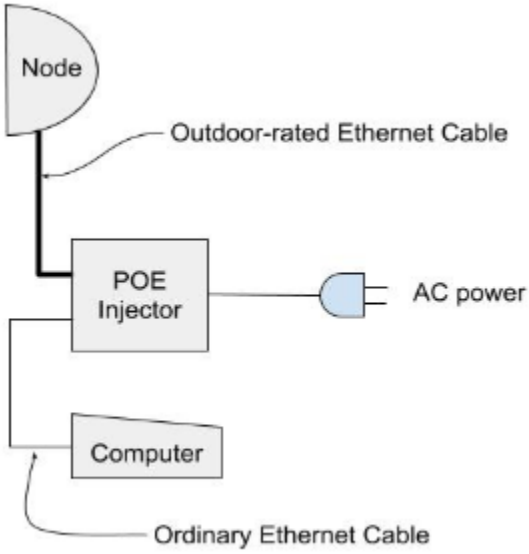


Fig. 1: Figure 1: Basic Installation

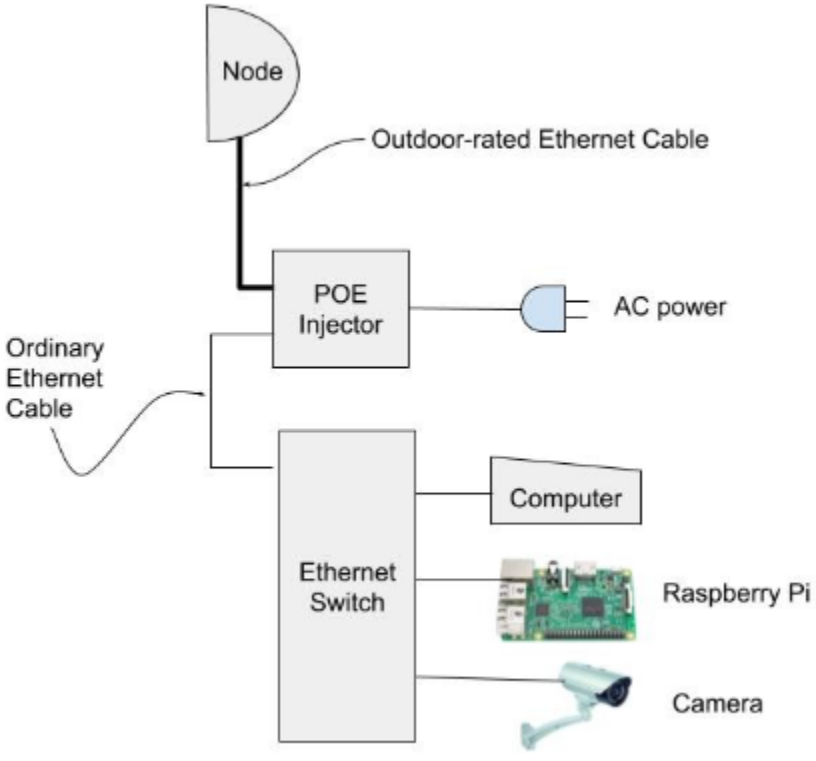


Fig. 2: Figure 2: Basic Installation with added Ethernet switch

CHAPTER

TWENTYSIX

TIPS FOR HANDLING FIRMWARE

Uploading firmware to an AREDN® node is usually a straightforward process. Follow the procedures documented in the **Downloading AREDN® Firmware** section to ensure you have the correct firmware version from the AREDN® website to install on your node. If you experience issues uploading firmware, the following tips may be helpful.

Web browser cache and sessions

One issue can occur when installing firmware using a web browser. Your computer's browser cache stores data for the URLs that have been visited, but IP addresses and other parameters may change during the install process. It is possible for the cache to contain information that doesn't match the latest settings for the URL, so the browser may block the connection. Clearing your computer's web browser cache will allow the latest URL settings to be registered so you can continue with the install process. During AREDN® firmware installs you can usually resolve a the issue with one or more of the following things:

- Refresh or Reload the URL for your node.
- Clear your browser cache and delete cookies.
- Close your browser and restart a new session.
- Use a different web browser program or a *Safe Mode / Incognito* browser window.
- Unplug and reconnect the Ethernet cable from your computer to ensure that your machine has received a new DHCP IP address on the same subnet as the node's updated IP.

PXE Server

If you are using a PXE server to provide your device with an IP address and a new firmware image, be sure to allow the PXE server through your computer's firewall. If the PXE server does not display any activity when you begin your firmware install, check your firewall settings. On the Windows control panel, for example, click *Advanced Settings* and look through the "Inbound Rules" to see if a rule exists for the PXE server. If a rule exists, make sure to "allow connection" for both private and public networks. If no rule exists, create a new rule allowing connection for both public and private networks.

26.1 Tips for Upgrading Firmware

Upgrading an AREDN® node is accomplished on the *Firmware* page. Follow the procedures documented in the **Downloading AREDN® Firmware** section to ensure you have the correct firmware version from the AREDN® website to install on your node. In rare cases the upgrade process may fail due to lack of node resources, but such a failure will leave the node running its previous firmware version.

Try to Sideload Firmware

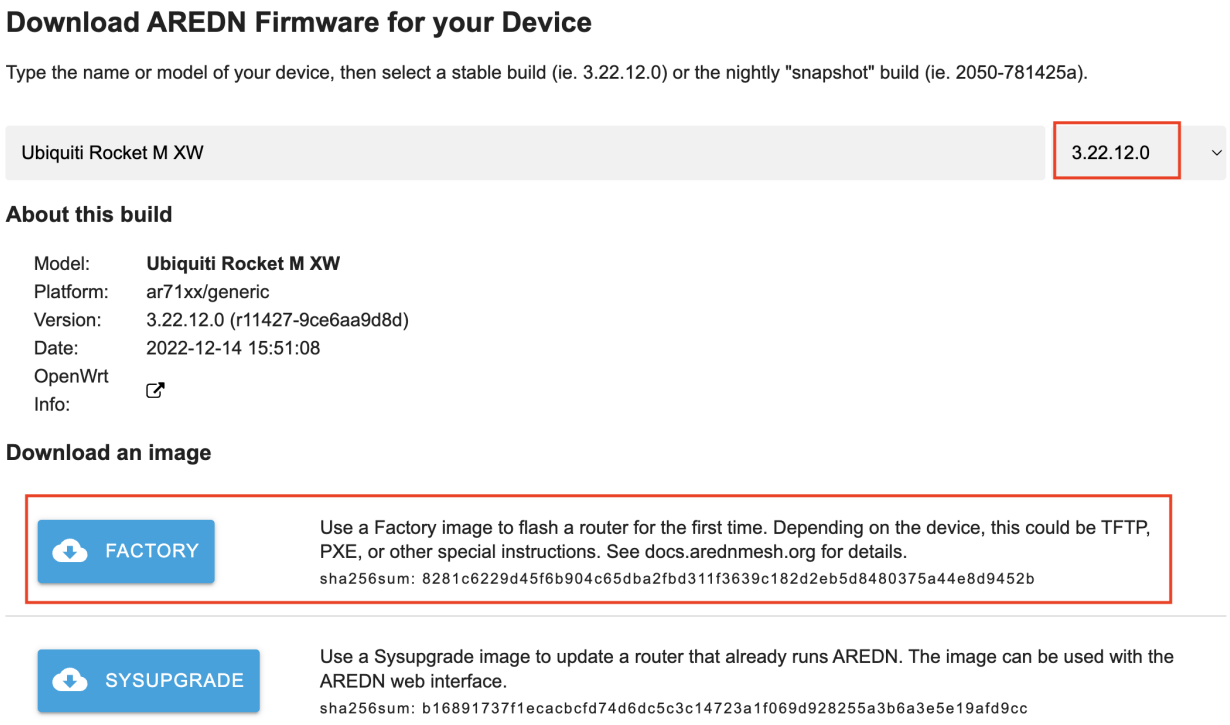
The **Sideload Firmware** option is described in the *Node Admin* guide. This involves using a file copy utility on your computer to copy the firmware file to a specific directory and filename on your node. Once the new firmware file is available on the node, you can click the *Update* button to start the install process.

> **Attention:** If you try to initiate a firmware install from your node's command line, do not use the legacy OpenWRT `sysupgrade` utility. That no longer accomplishes all of the correct steps to flash an AREDN® node. Instead you must use `/usr/local/bin/aredn_sysupgrade` to install a local firmware image from your node's command line.

26.2 Tips for Downgrading Firmware

Downgrading AREDN® firmware is typically accomplished using the same procedure as for uploading firmware to your node. You are simply uploading a previous version of the firmware rather than the latest version. If you are downgrading firmware on a node which previously used a different target architecture (`ar71xx` vs. `ath79`), you will need to do a fresh *First Install* using the appropriate firmware for that device.

Use the AREDN® Firmware Selector to download the previous release's install files. For example, if your Ubiquiti Rocket M5 XW is currently running version `3.23.4.0`, then download the files required for release `3.22.12.0` (as shown below).



To do a fresh install of the firmware, review the **Installing AREDN® Firmware** documentation and follow the steps for the install procedure that is appropriate for your node model.

- For Ubiquiti and TP-LINK models you will be uploading the *FACTORY* firmware.
- For Mikrotik models you will boot using the *KERNEL* file (which you rename to *rb.elf*) and then immediately apply the *SYSUPGRADE* firmware image.
- For GL.iNet devices you will use the recovery procedure to upload the *SYSUPGRADE* firmware image.

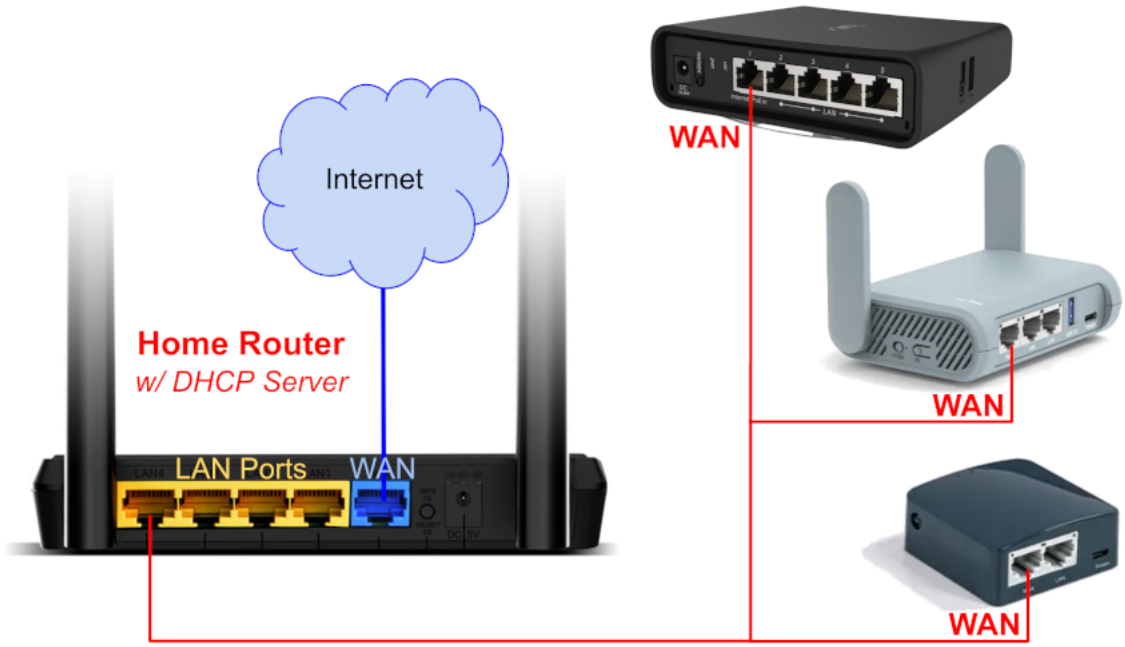
After downgrading your node's firmware you will then continue the process for entering your call-sign and configuring the node's settings.

CHAPTER

TWENTYSEVEN

CONNECTING NODES TO HOME ROUTERS

There are several indoor AREDN® nodes that have more than one Ethernet port. The AREDN® firmware running on these types of nodes has the WAN port preconfigured for connecting to the Internet. You can get the latest information about the specific port configured as the node's WAN port from the AREDN® website here: Ethernet Port Usage. It is recommended that you use a label maker to clearly identify the ports on your multiport devices.



When you connect the node's WAN port to one of the LAN ports on your home router, the node's WAN should receive an IP address on your home network from the router's DHCP server. Alternatively you can reserve an IP address in your home network range and assign the static IP to the node's WAN through the **Basic Settings** page on your node. There are many sources of information about basic home networking which will not be duplicated here, but feel free to familiarize yourself with IP networking through reading and research.

Once you have connected your node to your home router, Internet access will be available to the node itself as well as to any of the devices connected to the node's LAN network. It is not recommended

to allow Internet access through your node from mesh nodes. You can verify that mesh devices are not allowed to use your node's Internet (WAN) by disabling "Mesh to WAN" under the *Advanced Options* in the **Network** admin section.

CHAPTER

TWENTYEIGHT

POWER OVER ETHERNET (POE)

The phrase Power over Ethernet (PoE) encompasses any of several different standards and methods for passing DC power over twisted-pair Ethernet cabling. The advantage of PoE is that it allows a single cable to carry both data and power to your devices, and several AREDN® supported devices can be powered using PoE.

This section of the documentation provides a high-level overview for those who are not already familiar with this concept. You do not need to be an expert in *Power over Ethernet* technology, but it may help to be aware of a few concepts in case you run into these terms when researching PoE switches or injectors.

28.1 Passive PoE

At the present time, all of the PoE radios supported by AREDN® require the use of **Passive PoE**. In a *Passive PoE* system the power source does not negotiate voltage or wattage requirements with the powered device. *Passive PoE* power sources simply supply a specific voltage constantly, up to the maximum current limit that the power source allows.

The primary message of this section is to encourage you to read the manufacturer's data sheet carefully for the hardware that you will be deploying. Pay particular attention to the specifications for **Input Voltage** and **Maximum Power Consumption**. The allowed voltage ranges and maximum power consumption for AREDN® radios will vary by hardware model as shown in the comparison below.

Example Data Sheet Info

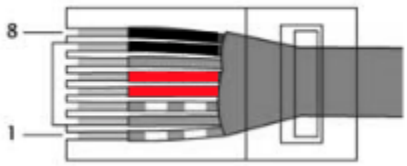
	Mikrotik LHG-5nD	Ubiquiti PowerBeam-M5-400
Input Voltage	11 - 30 vdc Passive PoE	11 - 28 vdc Passive PoE
Power Consumption	6W	6W

You simply need to determine what voltage range your equipment accepts and then use a power source that constantly supplies a voltage within that range. For example, the Mikrotik device in

the table above can accept a range between 11 and 30 volts, while Ubiquiti devices typically accept between 11 and 28 volts. This means that you could use either a 12v or 24v source to power these devices because both are within the acceptable voltage range for these radios.

In this example, both models have a *Maximum Power Consumption* of 6W. This means that you can expect a maximum current draw of approximately 500mA if you use a 12v battery to power them. If you use a 24v source then you would expect a maximum current draw of only 250mA.

Passive PoE commonly uses the same Ethernet cable wires as are used in the IEEE 802.3af Mode B standard, but that is the only similarity between *Passive PoE* and that standard. *Passive PoE* uses separate wire pairs to carry data or power as shown below. DC positive is carried on pins 4-5, DC negative is carried on pins 7-8, while data is carried on pins 1-2 and 3-6.



Cable Pins	Wire Pair Usage	
	Data	Power
Pin 1	RX+	
Pin 2	RX-	
Pin 3	TX+	
Pin 4		DC+
Pin 5		DC+
Pin 6	TX-	
Pin 7		DC-
Pin 8		DC-

You should not need to concern yourself with the various IEEE 802.3 standards that may be used for other types of PoE equipment. The radios currently supported by AREDN® do not use standards such as *802.3af, 802.3at, 802.3bt, PoE+, 4PPoE, or Ultra PoE*. There is a wealth of information on the Internet if you decide to learn more about these other standards.

Be aware that it should not damage your AREDN® device if you connect it to an 802.3af/at switch or PSE (power sourcing equipment). The only consequence would be that the device will not be powered, since switches using the other standards will not send power if they do not detect a compatible device.

CHAPTER

TWENTYNINE

COMMAND LINE ACCESS TO YOUR NODE

There may be times when it would be useful to have command line access to your node. AREDN® nodes support both Secure Shell (ssh) and Telnet. Both access methods will require a set of login credentials (*root* username & password). Linux and MacOS computers have native tools for both *SSH* and *Telnet*.

The *OpenSSH* package can be enabled on Windows computers. Use a web search engine to find information for your specific operating system (for example search "openssh for windows 10"). Here are some examples for enabling OpenSSH on Windows computers:

- Example for Windows 10
- Example for Windows 11

On Windows computers you can also use a terminal program such as PuTTY to connect to your node via ssh or telnet. To learn how to use these programs on your computer, please see the appropriate documentation for the specific programs you have chosen.

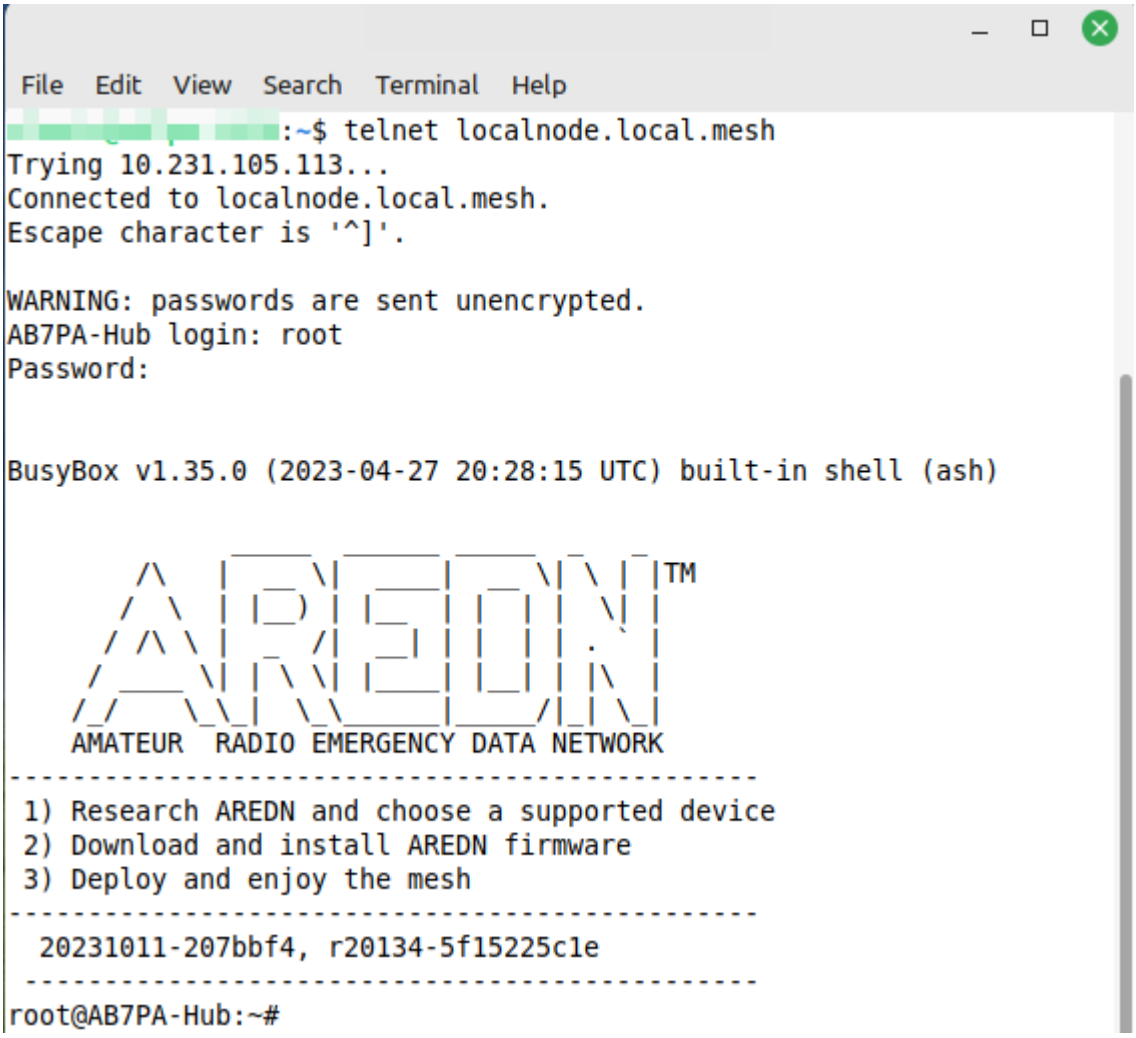
As shown in the command line examples below, you begin by opening a terminal window on your computer. At your computer's command prompt, enter the command string you will use to authenticate to your node.

Telnet

Telnet will prompt you for the *root* username and password before displaying your node's command prompt. The *telnet* protocol uses well-known port `23` and all traffic is unencrypted. An example *Telnet* command string is

```
$ telnet localnode.local.mesh
```

After successfully authenticating, your node's command prompt will be displayed.

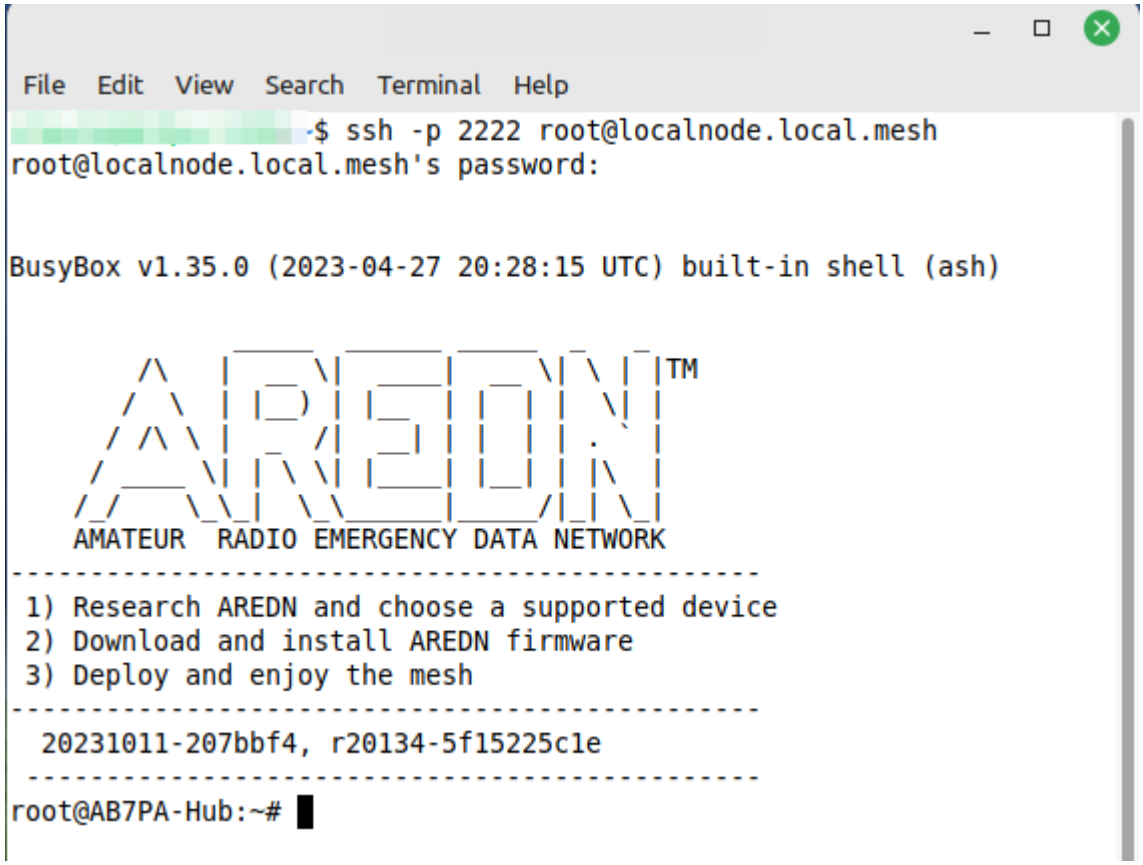


SSH

SSH requires you to specify the port, *root* username, and password on the command line. This is because AREDN® nodes do not use the default well-known *ssh* port [22], but instead use port `2222` for *ssh* connections. An example *SSH* command string is

```
$ ssh -p 2222 root@localnode.local.mesh
```

After successfully authenticating, your node's command prompt will be displayed.


File Edit View Search Terminal Help
-$ ssh -p 2222 root@localnode.local.mesh
root@localnode.local.mesh's password:
BusyBox v1.35.0 (2023-04-27 20:28:15 UTC) built-in shell (ash)
AMATEUR RADIO EMERGENCY DATA NETWORK
1) Research AREDN and choose a supported device
2) Download and install AREDN firmware
3) Deploy and enjoy the mesh
20231011-207bbf4, r20134-5f15225c1e
root@AB7PA-Hub:~#

CHAPTER

THIRTY

TIPS FOR AIMING DIRECTIONAL ANTENNAS

Contributor: Brett Popovich KG7GDB

AREDN® nodes with directional antennas can be challenging to align, especially if they have very narrow beam widths. The goal is to achieve the closest alignment in order to pass RF signals efficiently.

30.1 Practice with Nearby Nodes

If you can drive to within 1/4 mile of an active node, you should be able to pass signals well. At close range the aiming may not be as critical and you could even place a NanoStation or SXTsq panel on your dashboard. Find a public park, open parking lot, or street parking where you have line of sight to a remote node that uses the same frequency as your portable node. Here are some steps you can follow to practice aiming your node.

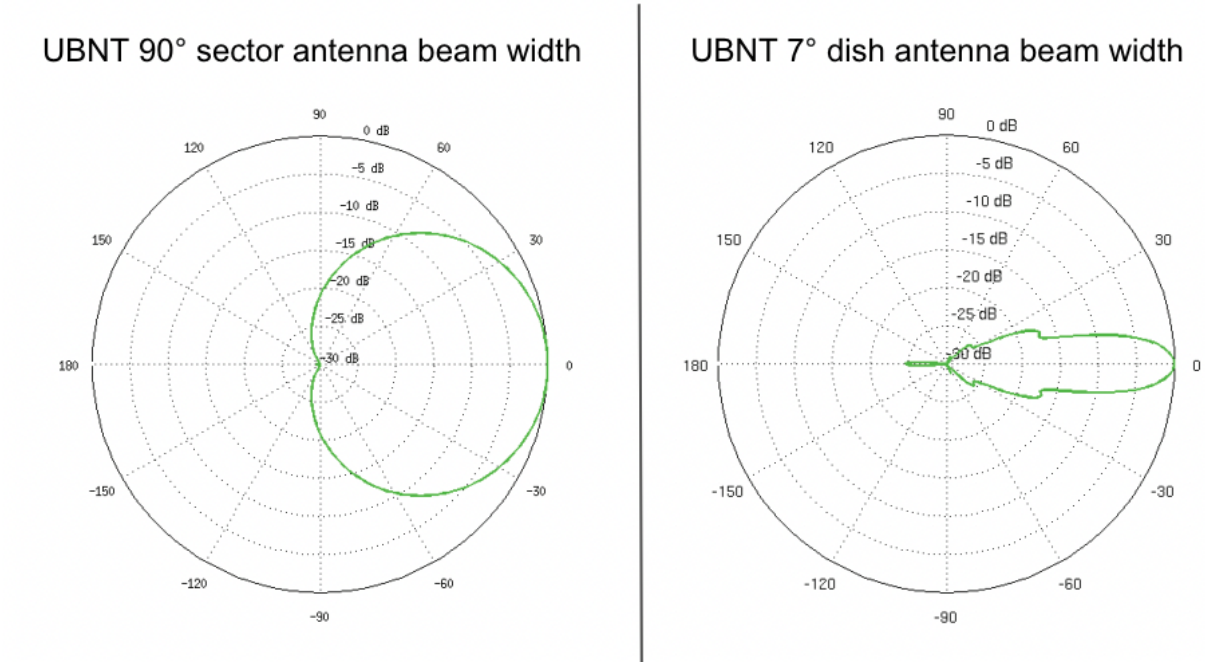
- In your vehicle, power up your node and plug in your laptop. Disable the wifi interface so the laptop gets its IP address from the node. Open a web browser and use `http://localnode.local.mesh` to load your node's home page. You will need to have your admin password to authenticate to *admin* mode.
- On the **Radio** page enter the SSID, channel, and channel width that matches the remote node you are surveying. If you changed any of these settings, click `Done` and `Commit` your changes.
- Now you can click the *Tools* icon to select **WiFi Signal** from the menu. Choose your remote node by clicking in the field at the right and selecting the desired remote node from the dropdown list. The signal level will be updated on the graph, and you can also enable an audio tone to give you an audible indication of the signal level. Turn your radio slowly or even change the car position to find the position at which the signal level is best.
- Once you have the highest SNR at your test location, click `Done` to return to the **node status** display. You should see the remote node in the list of *Neighborhood Nodes* in the center column. There will also be values displayed for the link quality. Hover over the row for the remote node and click to show the **Neighborhood Device** display. This will provide a wealth of information about the quality of the link to the remote node being tested.

- If desired you can click the neighbor node name to show the remote node's view of your connection by following the same procedure as above.

30.2 Aligning Distant Nodes

Distant fixed nodes can be aligned with the same tools you used in the previous section. Different antennas will have different beam widths depending on the model. Check the manufacturer specifications to determine the beam width of your antennas. This will give you a clue as to how precise your aim should be in order to send/receive signals effectively.

For example, Mikrotik LHG5 and Ubiquiti RocketDish5 antennas are very narrow, with beam widths between 5° and 7°. Mikrotik QRT panels and Ubiquiti Powerbeam antennas have beam widths between 10° and 12°. Mikrotik SXTsq5 panels and Ubiquiti AirGrid antennas have beam widths between 20° and 23°. Ubiquiti NanoStations and Mikrotik SXTsq2 panels have beam widths between 45° and 60°. Sector antennas have typical beam widths of 90° or 120°, while omnidirectional antennas cover 360° with various degrees of downtilt.



While it is helpful to know the antenna pattern for the nodes at both ends, the key is knowing the exact coordinates of the two locations so you can determine their topographical relationship to each other (horizontal and vertical azimuth). There are several computer tools for modeling radio links that were mentioned in the **Network Design Guide** under the *Network Modeling* section. One of

the most useful is VE2DBE's Radio Mobile which provides all of the required details for aiming directional antennas between two locations, including both true and magnetic bearings for both sides of the link.

Studying various local maps may allow you to discover other sites where you could place intermediate nodes that might link two distant locations. Google Earth can help you identify visible landmarks before aiming. Obvious tall objects such as water towers or multi-story buildings can be added as markers. Nearby objects such as church steeples or park features can be useful as visual reference points during the aiming procedure: for example, "I need to aim over the skate park to the left of the church to hit the remote node." Google Earth also provides a ruler tool which shows the bearing between map locations, and you can look at the Profile View to see whether there are features which may block your signal. Another tool mentioned in the **Network Design Guide** under the *Network Modeling* section is Radio Fresnel which generates a Google Earth KMZ file that identifies ground features which may block the Fresnel Zone along your link path.

Node 1		Node 2	
Latitude	33.39776°	Latitude	33.176596°
Longitude	-111.595515°	Longitude	-111.588652°
Ground Elevation	475.1 m	Ground Elevation	465.0 m
Antenna Height	12.0 m	Antenna Height	10.0 m
Azimuth	178.51 TN / 168.73 MG	Azimuth	358.52 TN / 348.76 MG
Tilt	-0.14°	Tilt	-0.08°

The chart above shows typical link details that are provided by Radio Mobile. It is very helpful to know these kinds of details and to have an accurate compass before you begin the antenna aiming process. If you use magnetic bearings you will need to know the declination for your location, and be sure your phone or compass is not influenced by nearby metal objects.

Some antennas are easier to aim than others. Large metal dishes are heavy and may require two people to aim, whereas lighter dishes like the Mikrotik LHG units are easy to manipulate. Often only a slight change in position can make a large difference in SNR and link quality. Be sure to avoid trees and be sure your link's first Fresnel Zone is clear of obstructions in order to achieve the best link quality. See the **Network Design Guide** on *Radio Spectrum Characteristics* for examples of ground clearance at different frequencies to ensure the Fresnel Zone is clear.

CHAPTER

THIRTYONE

THE BABEL ROUTING PROTOCOL

Contributor: Tim Wilkerson KN6PLV

The AREDN® team implemented a new routing protocol called Babel which replaced the original and obsolete protocol called OLSR. Babel has a number of qualities which make it good for AREDN®.

1. It's a loop free protocol so, regardless of how the network is changing, routing loops will never form in the network.
2. It's primarily a reactive protocol which sends changes to neighbors when needed rather than broadcasting its state continually.
3. The protocol understands the difference between wired, wireless, and tunneled links – the three link types AREDN® utilizes.
4. It's a layer-3 routing protocol, which integrates easily with how AREDN® already operates.
5. It's highly configurable which will allow an optimal setup for our use case.
6. Finally, it's simple.

If you're interested in more comparisons between Babel and other options, there are many good presentations on YouTube. This is a great primer on Babel itself.

Babel was initially deployed alongside OLSR with the two operating in parallel. This allowed us to examine Babel's performance in our networks without disturbing nodes which were not running Babel. The results indicated that Babel outperformed OLSR in several areas, providing a more stable network, better routing decisions, and lower network overhead.

CHAPTER

THIRTYTWO

SETTINGS FOR RADIO MOBILE

Contributor: Andre Hansen K6AH

Radio Mobile is a valuable timesaving tool for network planning and modeling. The results obtained depend upon the accuracy of the settings used to generate the model. The following Radio Mobile settings have proven useful.

Radio System Section	Recommended Setting
TX power (Watts)	0.25
TX line loss (dB)	0.5
TX antenna gain (dBi)	[varies]
RX antenna gain (dBi)	[varies]
RX line loss (dB)	0.5
RX threshold (V)	4

While the radio may have a TX Power specification of 1/2 watt (27 dBm), it's more accurate to use 1/4 watt (24 dBm) for dual chain (MIMO) devices because the power is split between the vertical and horizontal domains. The TX and RX Line Loss is minimal, so you can use 1/2 dB to account for the coax jumpers. Using 4 V for the Receive Threshold will approximate the device's receive sensitivity of -94 dB. It is usually best to underestimate the TX and RX Antenna Gain in order to obtain a more realistic model.

When Radio Mobile completes its link analysis, it will display the Fade Margin. For a solid connection a fade margin of 15 dB or greater is needed. Anything above that will only increase the MCS rate. For example, MCS15 requires 19 dB more received signal (94 - 75) and the Ubiquiti Rocket transmit power is 5 dB lower at that same rate, so you will need a total of 24 dB (19 + 5) additional fade margin (39 dB in total) to achieve that data rate. 39 dB is a large Fade Margin and is not often achieved on a link.

Determining the MCS Rate

If you telnet to your node, the following command will indicate the MCS rate the device is running:

```
cat /sys/kernel/debug/ieee80211/phy0/netdev:wlan0/stations/*/rc_stats
```

Here is an example from an endpoint node pointing to a backbone node over 25 miles away. The *Node Status* screen indicates -73/-95/22 dB SNR.

```
type          rate      throughput  ewma prob   this prob  retry   this␣
↪succ/attempt   success    attempts
HT20/LGI      MCS0            5.6      100.0       100.0      1          ␣
↪  0(  0)         1           1
HT20/LGI      MCS1           10.5      100.0       100.0      4          ␣
↪  0(  0)         4           4
HT20/LGI      MCS2           14.8      100.0       100.0      5          ␣
↪  0(  0)        93          93
HT20/LGI      MCS3           18.6       97.7       100.0      5          ␣
↪  0(  0)      1380        1416
HT20/LGI  tP  MCS4           25.1       99.9       100.0      5          ␣
↪  0(  0)     31688       33264
HT20/LGI      MCS5            8.6       25.8       100.0      0          ␣
↪  0(  0)       175        3495
HT20/LGI      MCS6            0.0        0.0         0.0      0          ␣
↪  0(  0)         1        3495
HT20/LGI      MCS7            0.0        0.0         0.0      0          ␣
↪  0(  0)         0        3495
HT20/LGI      MCS8           10.5      100.0       100.0      0          ␣
↪  0(  0)         1           1
HT20/LGI      MCS9           18.6       99.9       100.0      5          ␣
↪  0(  0)       368         380
HT20/LGI      MCS10          25.1       99.9       100.0      5          ␣
↪  0(  0)     37921       38776
HT20/LGI T    MCS11          30.3       99.9       100.0      5          ␣
↪  0(  0)    439091      448760
HT20/LGI      MCS12          14.1       33.2       100.0      6          ␣
↪  0(  0)      4482        8447
HT20/LGI      MCS13           0.0        0.0         0.0      0          ␣
↪  0(  0)         0        3495
HT20/LGI      MCS14           0.0        0.0         0.0      0          ␣
↪  0(  0)         0        3496
HT20/LGI      MCS15           0.0        0.0         0.0      0          ␣
↪  0(  0)         0        3495
```

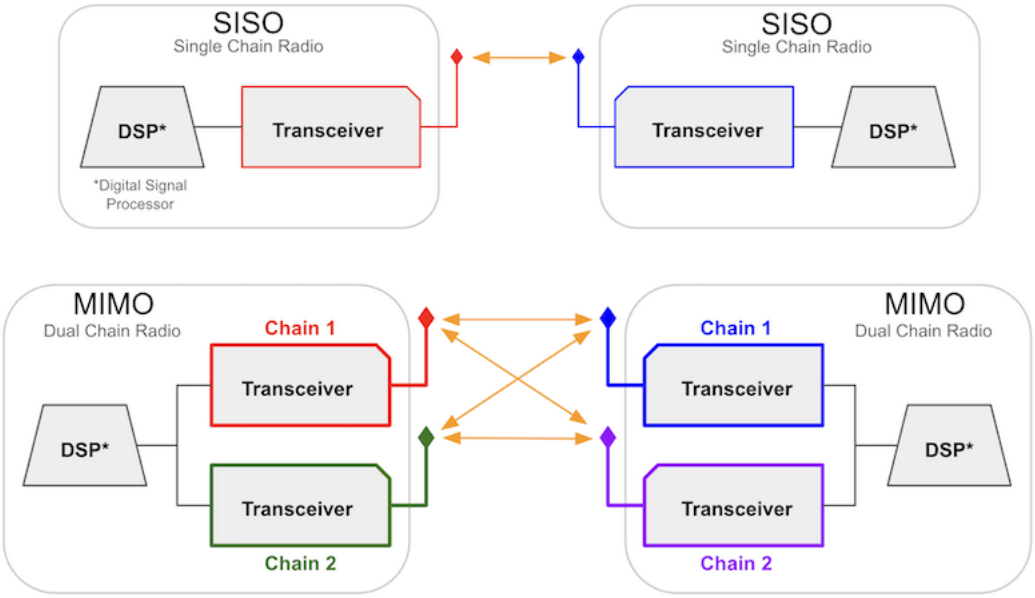
The "T" in the 10th character position indicates the current MCS rate, and a "t" indicates the current fallback rate. In this case the link is running MCS11 at 30.3 Mbps.

CHAPTER

THIRTYTHREE

COMPARING SISO AND MIMO HARDWARE

SISO (Single Input Single Output) device hardware has a single transceiver-antenna chain, while MIMO (Multiple Input Multiple Output) devices have multiple chains coordinated through the Digital Signal Processor (DSP). The MIMO devices supported by AREDN® have dual chains for both transmit and receive, and they support dual data streams [2x2:2].



Both SISO and MIMO devices use OFDM (Orthogonal Frequency Division Multiplexing), which inherently handles poor RF conditions such as multipath interference or fading. The rate selection algorithm in the wireless driver adapts to changing RF conditions so that the optimal MCS (Modulation and Coding Scheme) rate is always used. The selected MCS includes the appropriate modulation, forward error correction, and number of data streams.

33.1 SISO Device Hardware

By design SISO devices transmit all of their RF power on a single polarization. While it may seem like an advantage to have full power concentrated on a single polarization, there are specific limitations to SISO devices. A single chain device can only transmit one data stream at a time, and SISO devices do not have the ability to process and enhance multiple signals received simultaneously.

SISO devices are also limited in the data throughput they can achieve on their single chain. For example, a SISO device is limited to the 802.11n MCS7 (Modulation and Coding Scheme) protocol rate of 32.5 Mbps with Long Guard Interval (LGI) using a 10 MHz channel width, while a MIMO device using MCS15 (Modulation and Coding Scheme) can achieve up to 65 Mbps. In this regard SISO is at a definite disadvantage since it lacks sophisticated signal combining and the multiple simultaneous data streams that are possible with MIMO.

33.2 MIMO Device Hardware

One of the advantages of MIMO devices is their ability to exploit multipath signals, achieving a better Signal to Noise Ratio (SNR) by combining multiple received transmissions. This is accomplished using 802.11n technologies such as Polarization Diversity and Maximal Ratio Combining.

On MIMO devices the total transmit power is split between its two polarizations, which means that MIMO signals have lower EIRP per polarization. It is possible that SISO devices on both ends of a link could have SNR values that match those of MIMO devices using 802.11n MCSo (Modulation and Coding Scheme) to MCS7 on that same link. However, a MIMO device using MCS0 to MCS7 will transmit its data stream on both chains simultaneously, providing a distinct advantage on the receiving end where the MIMO device uses MRC to enhance the signal. MRC is used when multiple antennas receive the same data stream, which applies only for MCS0 to MCS7. With MCS8 to MCS15 Spatial Multiplexing achieves multiple simultaneous data streams.

Given the same channel width and link characteristics, MIMO tends to out-perform SISO in both reliability and throughput. A good test to verify this would be to compare the performance of SISO vs. MIMO between the same endpoints. MIMO can attain double the throughput because it is capable of using twice the MCS rate. In the final analysis, the technology limitations of SISO will not allow it to match the throughput levels that are possible with MIMO.

33.3 Troubleshooting Tips

- Whenever possible try not to mix device types on radio links. As a general rule, use MIMO-to-MIMO for most types of RF links.
- If you have a marginal SISO-to-SISO link and you must replace one of the radios, either install another SISO radio or replace both ends with MIMO devices. A marginal but usable link between SISO devices may become unusable if only one is replaced with a MIMO device.

Additional information on the operation of SISO and MIMO devices can be found in references such as this: MIMO for Dummies.

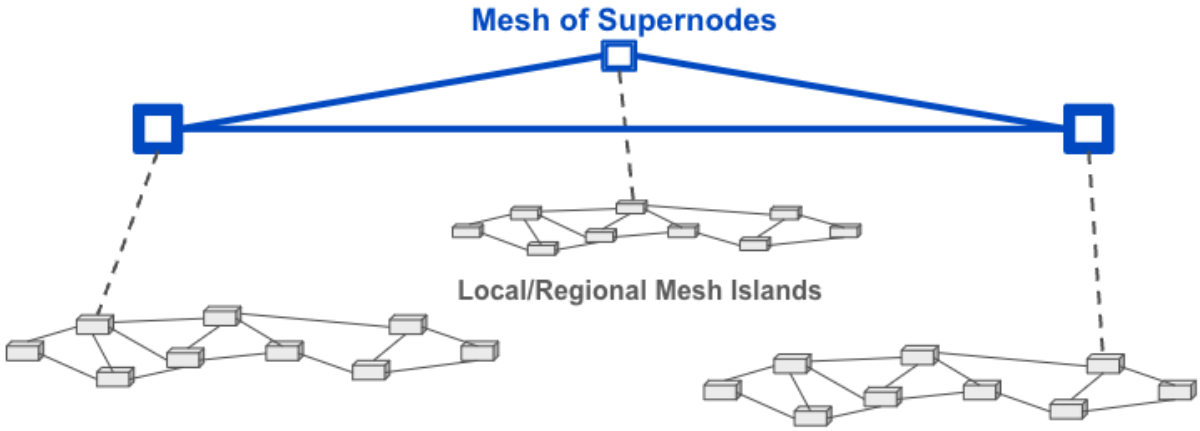
CHAPTER

THIRTYFOUR

CONFIGURING A SUPERNODE

Contributor: Tim Wilkinson KN6PLV

Supernodes are a way to link multiple mesh island networks in a safe and efficient way. A Supernode network is a high-level mesh network — **super** meaning *"above or higher."* The Supernode network sits above the individual mesh networks and provides connectivity without increasing the routing load on the local networks. Supernodes do not merge networks into one big mesh but instead isolate connections between discrete mesh networks. For further information see the *Supernode Architecture* section of the **Network Topologies** topic in the **Network Design Guide**.



34.1 Criteria for Deploying a Supernode

Before you consider deploying a Supernode, make sure you can adequately support the level of uptime that is desired for the Cloud Mesh network.

1. **Fast unlimited Internet connection.** Fiber is preferable. Low latency between Supernodes is important as is available bandwidth. A Supernode can easily transfer 1 teraByte of data every month, so an unmetered connection is best.

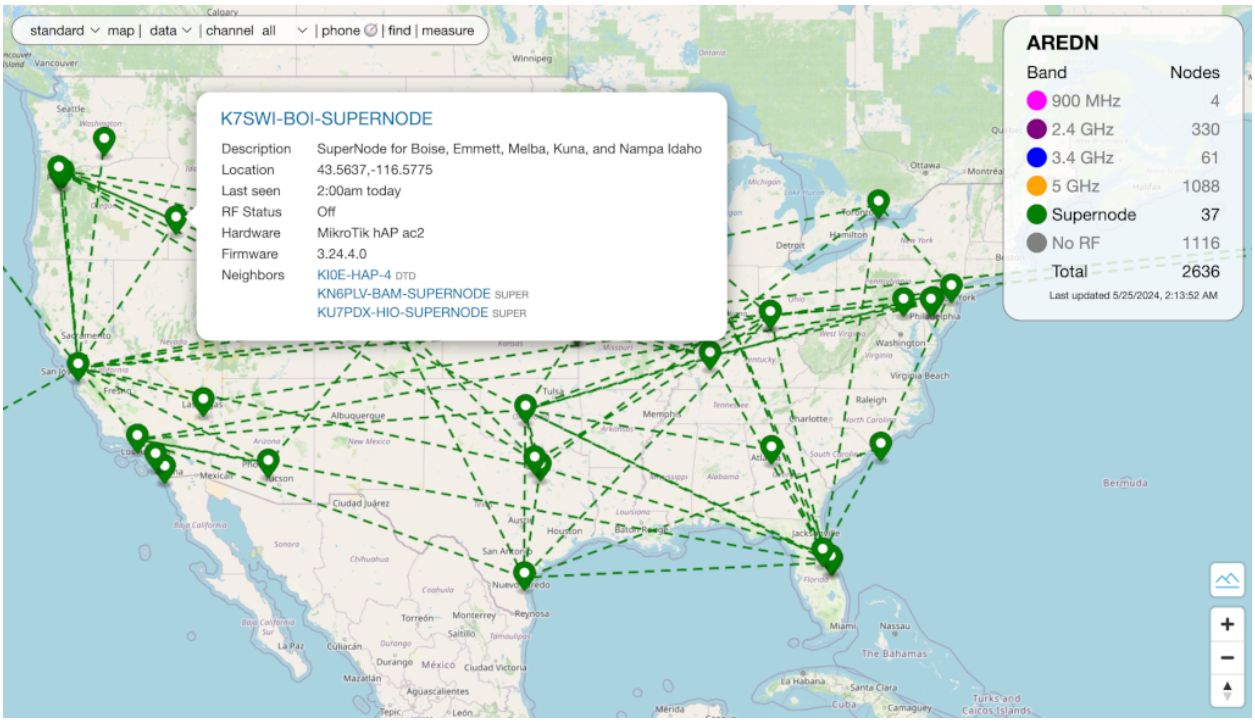
2. **Uptime stability.** The Supernode should be up 99.999% of the time. The location should ideally have backup power and network connectivity, both to the Internet and to the local mesh.
3. **Reliable local mesh connectivity.** As this will be the path for all traffic between your local mesh and every other mesh, the connection to your mesh should be at a high-bandwidth location. If you are deploying a Supernode with any sort of high-bandwidth backbone, the Supernode should be connected to the backbone.

34.2 Coordinating Supernode Deployments

As more Supernodes are deployed linking more local networks, the overall performance of the *Cloud Mesh* will be impacted. Therefore, you should coordinate the deployment of Supernodes among the Supernode owners at the time when tunnel links are requested for the *Cloud Mesh*. Your goal should be to choose **one** or **two** Supernode peers for your Supernode so that you can establish primary and backup links to the worldwide mesh. Having more than three or four peer Supernode links will only add unnecessary traffic to the entire system without providing actual benefit.

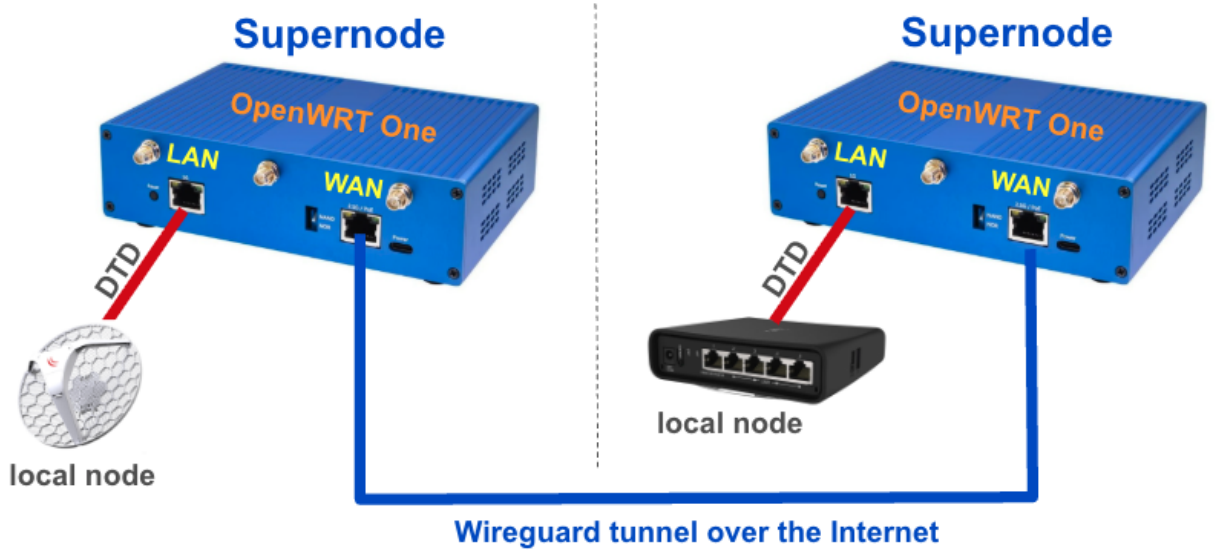
The number of messages a Supernode receives will scale linearly with the total number of nodes in all connected local networks. A Supernode receives a management message from every node in the network (all nodes in all local networks) every 5 seconds. With a typical message size of 100 bytes, a Supernode receives about 20 bytes per second per node. At the time of initial testing, there were 4,300 AREDN® nodes registered world-wide, so a Supernode for this network would receive `84 KB/s` or `0.7 Mb/s`, which is a manageable bandwidth requirement.

You may have multiple Supernodes on your local network, but each Supernode should only be connected to a single local network. By having each Supernode connected to only a single local network, the owners of each local network are responsible for their own Supernodes. This simplifies management and maintenance. There is also some fault isolation since a failed Supernode will only impact the one local network to which it is connected.



34.3 Setting up a Supernode

Typically a Supernode is configured on a dedicated **Cudy TR3000**, **OpenWRT One**, or **Virtual Machine**, although Supernodes can also run on *Mikrotik hAP ac2/ac3* hardware. Its sole task is to serve as a node on the Supernode network. The local network is linked to the Supernode using a DTD link on a LAN port. The Supernode is dedicated to this task, so it should not be used for anything beyond its role as a *Cloud Mesh* gateway.



The following steps are required to configure a Supernode.

1. Start with a device that is newly flashed with the latest Nightly Build. If the node has been previously configured or used beforehand, please reflash and start fresh in order to avoid problems later in the setup process.
2. Configure the Supernode with a nodename prefixed with your callsign followed by a location identifier as well as the word "SUPERNODE." For example you could use `AB2CD-NYC-SUPERNODE` or `AB6CD-LAX-SUPERNODE`
3. Ensure that the Supernode's radios are `off | disabled`
4. Provide a reserved or static IP address for the device's WAN connection to your Internet routing device.
5. Do not add any other configuration settings at this point or you may encounter problems later in this process. At this point you can `Commit` your changes and *reboot* the device.
6. Login to the rebooted device via *ssh* or *telnet* to get a command line prompt, and then manually type and execute each of these commands:

```
# uci -c /etc/config.mesh set aredn.@supernode[0].enable=1
# uci -c /etc/config.mesh commit aredn
# /usr/local/bin/node-setup
# reboot
```

Your node should now be functioning as a Supernode. To validate this you can do the following:

- Login to the Supernode via *ssh* or *telnet* and type the following command:

```
cat /etc/config/aredn
```

- Toward the end of the file which will be shown on the screen you should find the following lines:

```
config supernode
        option enable '1'
```

If you do not see these lines, please start this process again from the beginning and make sure to follow every step in the sequence.

Things to Avoid

Here are several things **NOT** to do when configuring your Supernode.

- Your Supernode must **not** use any cross-links (xlinks) to other nodes
- Your Supernode must **not** have tunnel links to any non-Supernode devices

- Your Supernode must **not** have its radio(s) enabled

Before proceeding, make sure all the previous steps have been completed successfully. Now you should be able to connect to another Supernode using a tunnel. The easiest way to do this is to ask another Supernode owner for a set of tunnel client credentials. Your node can use either a client or server tunnel link. Supernode owners can be identified from the Supernode Network Map

34.4 Configuring a Supernode Peer Tunnel

Supernode tunneling uses the Wireguard tunneling protocol, but the port range begins with port `6526`. On your Internet-connected router/firewall set the firewall rules to permit UDP traffic from the Internet on an appropriate range of ports. The starting port should be `6526`, which will provide for one supernode tunnel connection. If you want to allow up to 10 Supernode tunnel links (for example), then you would permit UDP traffic on the range of ports between `6526-6535`. Configure a port forwarding rule to send any traffic from the Internet on your range of ports to the IP address of your Supernode's WAN interface.

34.5 44Net and Supernodes

44Net addresses from ARDC can now be used within an AREDN® mesh for LAN devices. To allow these to be accessible across the supernode network, supernode will automatically route 44.0.0.0/9 and 44.128.0.0/10. However, if you are using 44Net address for other things in your network, this can cause problems. To disable this feature on your supernode do the following:

```
# uci -c /etc/config.mesh set aredn.@supernode[0].44net=0
# uci -c /etc/config.mesh commit aredn
# /usr/local/bin/node-setup
# reboot
```

CHAPTER

THIRTYFIVE

CREATING A LOCAL AREDN® SOFTWARE SERVER

There may be cases where your mesh nodes have no way to access the AREDN® servers for installing new software. One way to resolve this is to create your own software server on the local mesh and then point your nodes to this local service. The following sections describe the high-level tasks required to implement such a software service. In order to accomplish this, you may need to consult with someone who has System Administration skills for the specific platform you will be using to host your local software repository.

35.1 Configure your software server

Your software server must be connected to the mesh as a host on your local node's LAN network, using a node that also has Internet access via its WAN interface. The reason this node is connected to the Internet is to allow the web server to download updated files from the AREDN® downloads server. You should add this host to the node's *DHCP Reservation List*. You do not need to add the software host to the *Advertised Services List* of the node to which it is connected. The software server should be given a hostname that is unique on your mesh, typically prefixed with the callsign of the server owner. You can use any operating system platform you desire *(Windows, Linux, Mac)*, as long as it has the ability to function as a web server. The following are the two main tasks required of the local software server:

- Obtain the set of AREDN® software files from `downloads.arednmesh.org`
- Make those files available via its web server so nodes can query the software URLs

There are several ways to accomplish these tasks, and the best approach may vary depending on the platform you implement for your software server. Downloading the AREDN® software files can be done manually as needed, or the process could be automated and executed on a regular schedule. The recommended method is to use the rsync program which supports recursive copying of only the changed files on the source. An example *rsync* command is shown below:

```
/usr/bin/rsync -rv --delete --size-only downloads.arednmesh.org::aredn_
↪firmware /var/www/html/
```

Once you have a local duplicate of the AREDN® files, you need to verify that your copy of the files have the correct path pointing to your local software repository. In the example above, the local repository was placed in /var/www/html/, so you will navigate to that directory.

Under this directory you should see the `afs` subdirectory which contains all of the information used by the AREDN® Firmware Selector. Navigate into the `afs/www` subdirectory and use the editor of your choice to edit the `config.js` file. Locate the *Image download URL* section and change the default value of the *image_url* variable to point to your local download server. The default value is `image_url: "http://downloads.arednmesh.org"` and the example below shows the line edited to point to an example server.

```
image_url: "http://my-software-server.local.mesh",
```

Once the `config.js` file has the correct local URL, you will write the new path into all of the firmware selection entries. You accomplish this by running a Python3 script that ensures all of the files receive the current configuration settings from `config.js`. Navigate to the top level directory where you stored your copy of the firmware repository (in our example the local repository was placed in /var/www/html) and run the `collect.py` script to update the local URL path. It resides in the `afs/misc` directory and requires two arguments.

1. the path to the top level directory where you stored the arednmesh files (in our example: /var/www/html)
2. the path to the AFS www directory (in our example: /var/www/html/afs/www)

If you are already in that directory, you can use relative paths as in the example below.

```
cd /var/www/html/
./afs/misc/collect.py  .  ./afs/www/
```

Now your local AREDN® software source is configured to serve its files to any local nodes which want to update their firmware from your repository.

Next make these files available to network nodes via your web server. The steps for accomplishing this task will vary based on the specific web server software you are using. For example, using the Apache Web Server, you could store the software files under the web server's *DocumentRoot* (as in the example above) or you could create an *Alias* to allow web access to parts of the filesystem that are not under the Apache *DocumentRoot* (as described here). Once the software has been made available via the web server, you should be able to enter that URL in a web browser to see the entire software tree as shown in the example below.

Name	Last modified	Size	Description
Parent Directory		-	
afs/	2024-05-03 20:40	-	
firmware/	2024-05-05 20:48	-	
messages/	2024-06-28 13:54	-	
releases/	2024-07-03 07:00	-	
snapshots/	2024-07-03 06:56	-	

35.2 Point nodes to the local server

To point a node to the local software repository while in *admin* mode, navigate to the **Firmware** section and click on *Advanced Options*. The default Firmware URL is `http://downloads.arednmesh.org` but you can change this to the URL of your local software server. It is good practice to use the fully qualified domain name (FQDN) so the node will be able to resolve the domain portion of the URL to the mesh host's IP address. The URL you enter should match the alias or path you created and tested on your web server as described in the previous section and shown in the example below.

When you are finished with your changes, click the `Done` button. You will then be returned to your node's *admin* view where you will be able to `Commit` or `Revert` your changes. Once the node has been pointed to the local software server, you can navigate to the **Firmware** section and click the *refresh* icon to select and install the available software from your local software repository.

CHAPTER

THIRTYSIX

USING XLINKS

Contributor: Tim Wilkinson KN6PLV

A cross-link (xlink) allows you to pass AREDN® traffic across non-AREDN® network links. Tunnels and xlinks both connect two nodes together, so they are the same in that respect. However, they do it in very different ways.

Tunnels

Tunnels are a simple to use, all in one feature, which operates over your regular Internet to connect two AREDN® nodes. There is a bit of configuration information to exchange, but it is all fairly easy to set up. Tunnels *only work* over your **WAN** connection, you use the IP address given by the server, and there is very little else to configure.

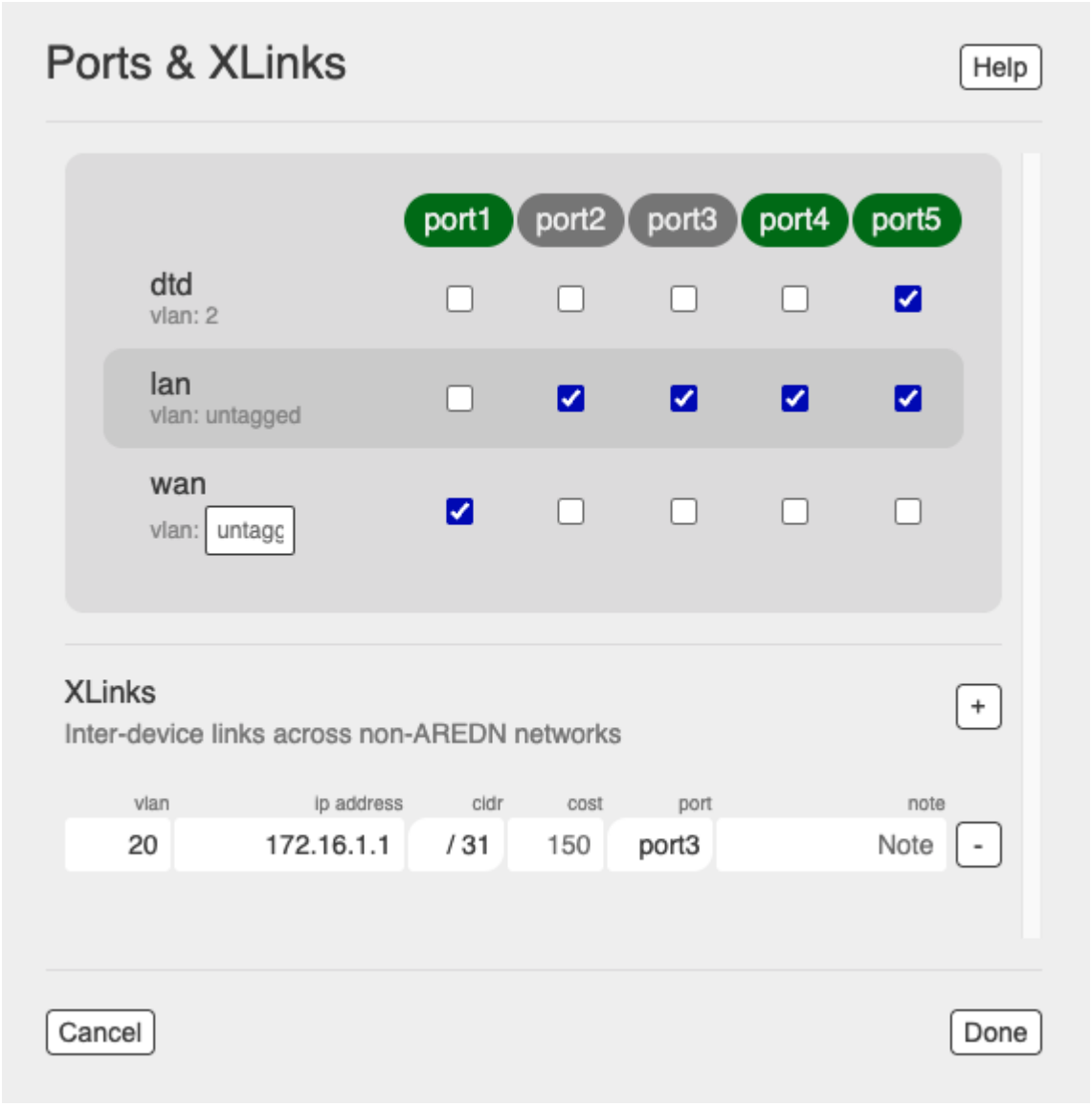
Xlinks

Xlinks, on the other hand, are much more basic and flexible. The configuration lets you choose IP addresses yourself, as well as setting a VLAN and *port* on which xlink traffic leaves the device. The IP addresses let the system route the data, but unlike a tunnel you can set these addresses any way you desire. You choose any unused VLAN number yourself, and the *port* determines how you want the data to be physically sent into or out of the node. How the data is moved to the peer device is not defined in any way, and deliberately so. Maybe you want to connect that *port* directly to a non-AREDN® PtP radio. Maybe you feed it into a switch then use some other tunneling technology to get it where it needs to go. Maybe it is just a bit of Ethernet cable. It is entirely up to you. Personally, I use tunnels to connect nodes over the Internet, but I use xlinks to connect nodes over Point-to-Point radios which are not running AREDN® firmware.

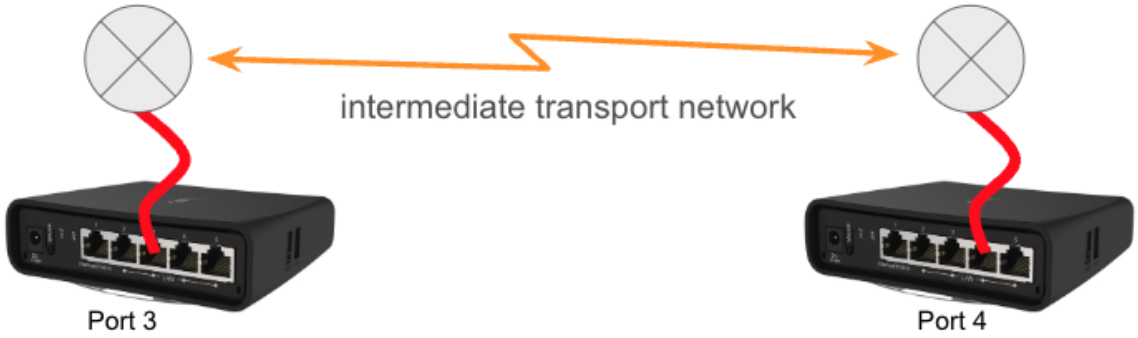
36.1 Configure the AREDN® nodes at both ends

In this example a *Mikrotik hAP ac2* was used as the AREDN® device on each end of the xlink. Navigate to the **Ethernet Ports & Xlinks** page of the node on one side of the link. To add an xlink click the *plus* icon, enter an unused VLAN number for the link, the IP address for the link, the CIDR netmask, a weighting factor, the port to which the near-side device is connected on your node, and .

The *cost* will be used by the routing protocol to determine the best route for AREDN® traffic. You may also add a note to describe the link or provide contact information.



In this example VLAN `20` is not in use anywhere else on the network. We assigned an *IP Address* of `172.16.1.1` with a netmask of /31. The xlink knows nothing about the details or configuration of the intermediate transport devices. The *cost* is set to `150` which is the same weight as would be used by a tunnel connection, but this can be increased if you want the cross-link to be chosen at a lower priority for routing traffic on the mesh. `port3` was chosen because it is an open port on this node. After entering these values, click `Done` and `Commit` your changes. Now you can cable your near-side transport device to port 3 on your AREDN® node.



36.2 Configure the intermediate transport link

How data is moved between the peer devices is not restricted or defined. There are many types of intermediate transport products that can be used to establish an AREDN® xlink. Refer to your manufacturer's documentation for the best way to ensure that network packets can be successfully transferred between the two endpoint devices. The easiest way to accomplish this is to bridge the traffic directly between the peer devices.

CHAPTER

THIRTYSEVEN

CUSTOM APP LAUNCHER

There may be times when you want to create your own customer application to run on your node, and you would like to include a custom launcher icon in the left nav bar of the web interface. This can be accomplished using the following steps:

1. Create a subdirectory tree under your node's `/www/` directory to store your application executable. In the example below the name of the application is `Performance-Snapshot`, so the directory path will be:

```
/www/cgi-bin/apps/Performance-Snapshot/
```

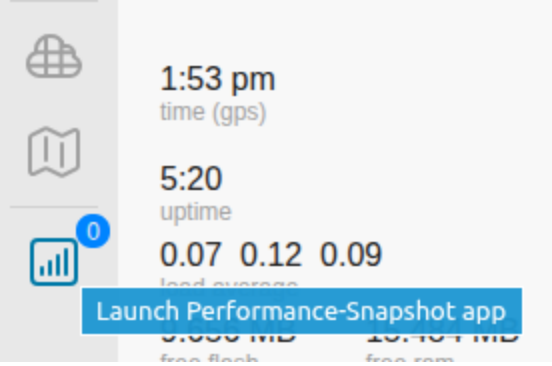
Copy your application executable into this directory and change the filename to `user` if you want the launcher to be visible to everyone. If you only want the launcher to be visible when logged in, change the application's filename to `admin`.

2. Create a subdirectory tree under your node's `/www/` directory to store your application's launcher icon (SVG format only). Since the name of the application is `Performance-Snapshot`, the directory path will be:

```
/www/apps/Performance-Snapshot/
```

Copy your application icon into this directory and change the filename to `icon.svg`.

3. At this point you can reboot your node or simply restart `uhttpd` on your node. The new application launcher icon should be visible in the left nav bar, and clicking that icon should open a new browser window or tab for your application to run in.



37.1 Adding an app badge counter and badge-color

You may be implementing an app that is capable of displaying an app badge, for example, a message count on the app icon (as shown in the image above). To display the app counter on the icon, your app must update the `badge` file with the appropriate value or count.

1. Create a new subdirectory on your node for the badge data. In this example the name of the application is `Performance-Snapshot`, so the directory path will be:

```
/tmp/apps/Performance-Snapshot/
```

2. Once this directory exists, create a `badge` file which your app will update with the latest value or count. In this example, the current value is `3`.

```
cat /tmp/apps/Performance-Snapshot/badge
3
```

3. If you want to set the color of the badge, you can add a `badge-color` file containing the hex color value for displaying the badge. In this example the current badge color value is `#1E90FF` (as shown below).

```
cat /tmp/apps/Performance-Snapshot/badge-color
#1E90FF
```

CHAPTER

THIRTYEIGHT

VIRTUAL MACHINE INSTALLS

Contributors: Trevor Raty KG6MDW and Tim Wilkinson KN6PLV

The use of virtual machines as AREDN® nodes is for advanced users. Most users should use *Mikrotik ac2* or *ac3* hardware to achieve similar functionality. These instructions are provided with the assumption that you understand your virtualization platform and are familiar with creating images and uploading virtual disks. The x86_64 image has been tested and is considered stable on the Proxmox, Unraid, QEMU, and VMware ESXi platforms, so usage on other virtualization platforms may not work as expected.

38.1 Prerequisites / Image information

At a minimum the VM must have two virtual CPUs, 64mb memory, and 128mb of storage. Providing more CPU is generally not needed on modern hardware.

There are two modes for networking: single-port and multi-port. Set the number of interfaces *before* powering on the VM for the first time. Regardless of the number of interfaced created, the node will initially be configured in 'single-port mode'. This can be changed later in the AREDN UI.

Single-port mode
: All traffic utilizes VLANs as described in the *Advanced Options* section of the *Network Settings* dialog in the **Node Admin** documentation. This requires your virtual interface to be VLAN aware or to be set as a passthrough interface.

Multi-port mode
: Ports can be assigned as needed to be LAN, DtD or WAN links. If your virtual interface is VLAN aware, you can tag VLANs; otherwise the interface should be untagged, which is the recommended setting. As an example, if you have three interfaces defined, you might assign ports as follows:

  - First interface: WAN
  - Second interface: DtD

- Third interface: LAN

Note: The images do not include any *vmtools* but they do contain drivers for the standard QEMU/VMware paravirtualized storage and networking. Using the paravirtualized devices is recommended.

38.2 Proxmox Installs

Proxmox Virtual Environment is an open-source server management platform for virtualization. There is an updated checklist of steps for Proxmox installs on the Bay Area Mesh Wiki.

Additionally, for a more in-depth example install utiltizing pre-configured VLANs, check out this RogueSecurity blog post.

38.3 QEMU Installs

1. Download the latest firmware image from the AREDN® downloads website.
2. Extract the .gz file. *7zip* on Windows may have issues with the .gz file, so you may need to download *gzip* for Windows or extract it on a Linux or Mac computer/VM.
3. Upload/copy the `.img` file to your VM server. You can rename the image if you desire.
4. Create the VM/Domain on your server and assign the `.img` file to it.
5. Boot the VM and proceed with the AREDN® node configuration steps.

38.4 VMware Installs

For VMware you will need to use QEMU tools or another V2V converter in order to convert the image to `vmdk` format. Some example software is listed below:

- QEMU for Windows binaries (Unoffical)
- QEMU Official downloads
- Starwind Converter

1. Download the latest firmware image from the AREDN® downloads website.
2. Extract the .gz file. *7zip* on Windows may have issues with the .gz file, so you may need to download *gzip* for Windows or extract it on a Linux or Mac computer/VM.

3. Convert the `.img` to `.vmdk` using your V2V converter of choice. For example, if you are using QEMU, open a terminal/command prompt and on Windows navigate to where QEMU is installed (normally `c:\Program Files\qemu\`). Run the following command, replacing "aredn.vmdk" and "aredn.img" with the filenames you have chosen.

```
qemu-img convert -f raw -O vmdk aredn.img aredn.vmdk
```

If you are using Virtualbox, below is the built-in command, replacing "aredn.vmdk" and "aredn.img" with the filenames you have chosen.

```
VBoxManage internalcommands createrawvmdk -filename aredn.vmdk -rawdisk␣
↪aredn.img
```

4. Create the VM/Domain on your server, but *do not assign it a disk*.
5. Upload/copy the `.vmdk` file to your server. You can rename the image if you desire.
6. `ssh` to the ESXi host, navigate to where the `.vmdk` file was uploaded and run the following command to verify/fix any conversion issues. This step helps to identify and fix potential image errors.

```
vmkfstools -i uploaded.vmdk verified.vmdk
```

7. Assign the verified `.vmdk` disk to the VM.
8. Boot the VM and proceed with the AREDN® node configuration steps.

CHAPTER

THIRTYNINE

TOOLS FOR INTEGRATORS

This section of the AREDN® documentation contains information useful for people who want to retrieve information from one or more nodes for use in different applications. For example, an integrator may want to periodically poll a set of nodes to gather link quality or signal values to insert them into a network management or historian system for trending and analysis.

39.1 SYSINFO

The **sysinfo** API (Application Programming Interface) has been included in AREDN® firmware for quite some time, and each update includes an *api_version* tag which can be used to track the feature set supported by that version of the API. As new features are added, the *api_version* number is incremented.

The basic API retrieves general node information in JSON format, and it can be invoked using the following URL: `http://<nodename>.local.mesh/a/sysinfo`

Note: In previous releases the API was accessed at `http://<nodename>.local.mesh/cgi-bin/sysinfo.json` but in recent releases that URL will be redirected to `http://<nodename>.local.mesh/a/sysinfo`

The following information is always returned in the JSON data stream:

- Node name
- API version
- Latitude, longitude, and grid square (if available)
- *Node Details* section containing the firmware manufacturer and version, the radio model and board ID, WAN sharing status, and the node description text (if any)
- *Sysinfo* section containing node uptime and load averages for the last one, five, and fifteen minutes

- *Interfaces* section containing the name, MAC address, and IP address (if any) assigned to each of the node's network interfaces
- *Mesh* section containing the SSID, channel, center frequency, channel width, and status of the mesh radio
- *Tunnels* section showing whether the tunnel package is installed and the number of active tunnels (if any)

The values returned by the API are represented in the following snippet of raw JSON. This is only a sample of the full data stream containing all of the values described above.


```
{
"api_version": "2.0",
"node": "CALLSIGN-NAME",
"node_details": {
  "model": "MikroTik hAP ac2",
  "board_id": "MikroTik hAP ac2",
  "firmware_mfg": "AREDN",
  "firmware_version": "3.25.8.0",
  "mesh_gateway": false,
  "mesh_supernode": false
},
"tunnels": {
  "active_tunnel_count": 2
},
"lat": 33.xxx,
"lon": -111.xxx,
"gridsquare": "DM33xx",
"meshrf": {
  "status": "off"
},
"sysinfo": {
  "uptime": "0 days, 3:48",
  "loads": [0.15, 0.03, 0.01],
  "freememory": "52240"
}
```


In addition to the basic information described above, which is always returned with every invocation, the **sysinfo** API can also include other details based on the flags appended to the URL as explained below. In some cases it may be useful to include more than one of the following flags in the URL, and these flags can be combined using the & operator. For example, `sysinfo?hosts=1&services=1` will include both the *hosts* and *services* information in addition to the basic details which are always returned.

39.1.1 Add Hosts Information

To retrieve mesh hosts information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/a/sysinfo?hosts=1`

A *hosts* section will be included in the JSON data stream containing an entry for each node and mesh-connected device. The *name* and *IP* address of each device will be shown. The values returned by the *hosts* flag are represented in the following snippet of raw JSON.

```
...
"hosts": [
  {
    "name": "CALLSIGN-NODE-22",
    "ip": "10.22.22.22"
  },
  {
    "name": "CALLSIGN-VOIP-PHONE",
    "ip": "10.22.22.24"
  },
  {
    "name": "MYCALL-NODE-81",
    "ip": "10.81.81.81"
  },
  {
    "name": "MYCALL-RPI",
    "ip": "10.81.81.83"
  }
],
...
```

39.1.2 Add Services Information

To retrieve mesh services information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/a/sysinfo?services=1`

A *services* section will be included in the JSON data stream containing an entry for each service available on the mesh. Each entry will include the service *name*, *protocol*, and *link* URL. The values returned by the *services* flag are represented in the following snippet of raw JSON.

```
...
"services": [
  {
    "name": "IperfSpeed",
    "ip": "10.2.174.80",
```

(continues on next page)

(continued from previous page)

```
    "protocol": "tcp",
    "link": "http://MYCALL-NODE-81/iperfspeed"
  },
  {
    "name": "EtherPad",
    "ip": "10.2.174.81",
    "protocol": "tcp",
    "link": "http://MYCALL-RPI:9001/"
  },
  {
    "name": "MeshChat",
    "ip": "10.2.174.82",
    "protocol": "tcp",
    "link": "http://MYCALL-RPI/meshchat"
  }
],
...
```

39.1.3 Add Local Services Information

To retrieve information about the services provided only through a single node, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/a/sysinfo?services_local=1`

A *services_local* section will be included in the JSON data stream containing an entry for each service available through the node being queried. Each entry will include the service *name*, *protocol*, and *link* URL as described above.

39.1.4 Add Link Information

To retrieve mesh link information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/a/sysinfo?link_info=1`

A *link_info* section will be included in the JSON data stream containing an entry for each node that is reachable via RF, DTD (Device To Device), or WIREGUARD (tunnel) from the node being queried. Each entry will be identified by the IP address of the reachable node, and within each IP address section you will see the *hostname* (node name), *linkType* (RF, DTD, or TUN), *linkQuality*, *neighborLinkQuality*, *signal*, *noise*, *olsrInterface* name, *tx_rate*, and *rx_rate*. The values returned by the *link_info* flag are represented in the following snippet of raw JSON.

```
...
"link_info": {
```

(continues on next page)

(continued from previous page)

```
  "10.22.22.22": {
    "hostname": "CALLSIGN-NODE-22",
    "linkType": "RF",
    "linkQuality": 0.9543000000,
    "neighborLinkQuality": 0.9748576110,
    "signal": -76,
    "noise": -95,
    "olsrInterface": "wlan0",
    "tx_rate": 6,
    "rx_rate": 4
  },
  "10.81.106.77": {
    "hostname": "MYCALL-NODE-81",
    "linkType": "DTD",
    "linkQuality": 1,
    "neighborLinkQuality": 1,
    "olsrInterface": "eth0.2"
  }
},
...
```

39.1.5 Add LQM Information

To retrieve Link Quality Monitor information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/a/sysinfo?lqm=1`

An *lqm* section will be included in the JSON data stream showing the current LQM configuration settings as well as an entry for each node that is reachable via RF, DTD, or TUN (Tunnel) from the node being queried. Each entry will be identified by the MAC address of the reachable node, and a variety of parameters will be displayed showing the tracked status of each link. The values returned by the *lqm* flag are represented in the following snippet of raw JSON.

```
...
"lqm": {
  "enabled": true,
  "config": {
    "user_blocks": ""
  },
  "info": {
    "now": 14478,
    "trackers": {
      "00:00:ac:1f:68:30": {
```

(continues on next page)

(continued from previous page)


```
        "lastseen": 14478,
        "lastup": 38,
        "type": "Wireguard",
        "device": "wgs1",
        "mac": "00:00:ac:1f:68:30",
        "ipv6ll": "fe80::200:acff:fe1f:6830",
        "refresh": 15081,
        "lq": 100,
        "rxcost": 206,
        "txcost": 306,
        "rtt": 46,
        "tx_packets": 6561,
        "tx_fail": 0,
        "avg_tx_packets": 20.592814410677,
        "last_tx_packets": 6561,
        "babel_route_count": 75,
        "babel_metric": 344,
        "routable": true,
        "user_blocks": false,
        "babel_config": {
          "hello_interval": 4,
          "update_interval": 120,
          "rxcost": 206
        }
    "distance": 80550,
    "hidden_nodes": [
    ],
    "total_route_count": 125
    }
  }
...
```

CHAPTER

FORTY

CONTRIBUTING TO DOCUMENTATION

If you are interested in contributing to the rapidly growing set of AREDN® documentation you can easily do so on GitHub. To contribute to the AREDN® project you first must create your own GitHub account. This is free and easy to do by following these steps:

1. Open your web browser and navigate to the GitHub URL.
2. Click the `Sign Up` button and enter the required information. We suggest using your callsign as the username.
3. On the GitHub website, click the `Sign In` button and authenticate to GitHub with the credentials you created.
4. Navigate on GitHub to the AREDN® documentation repository: https://github.com/aredn/documentation.
5. Click the `Fork` button at the upper right corner of the page. After this process completes, you will have your own copy of the AREDN® documentation files on your GitHub account.
6. Go to your local computer and clone your fork of the AREDN® documentation: `git clone https://github.com/YOUR-GITHUB-ID/documentation`
7. Navigate on your local computer to the folder where your cloned copy of the repository is located: `cd documentation` This directory contains your local copy of the AREDN® documentation, and all of your document editing should be done while you are in this directory or its subdirectories.

The workflow for contributing documentation is described in the file titled How to Use GitHub for AREDN®, a copy of which you will have in your new local repository. Refer to that document for additional information about contributing AREDN® documentation.

Your local editing branch name can be anything that makes sense to you as you add topics to the documentation. AREDN® documentation is written using the reStructuredText markup language and your text is saved in "rst" files.

Note: Before committing your changes, be sure to test your `rst` files locally using Sphinx to ensure they will render correctly.

After you create a Pull Request on GitHub, the AREDN® team will review your changes. Once your documentation contributions are committed to the AREDN® GitHub repository, a webhook automatically updates and builds the latest docs for viewing and exporting on ReadTheDocs.org. All contributions that are included by the AREDN® team in the documentation set will be covered by the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license held by *Amateur Radio Emergency Data Network, Inc.*

CHAPTER

FORTYONE

RESPONSIBLE DISCLOSURE POLICY

The members of the AREDN® team believe in the responsible disclosure of security vulnerabilities that may be discovered in the software. To further the goal of responsible disclosure, we request those persons who believe they may have discovered a security vulnerability to contact the *Security Team* via email at: **securityteam@arednmesh.org** The *Security Team* will work with you to ensure that the vulnerabilities are patched prior to public disclosure.

Furthermore we understand that other organizations may be developing firmware based on the solutions we have published. To that end the AREDN® group has created a security program for such organizations to be informed of discovered vulnerabilities so they can secure their offerings prior to the public disclosure of such vulnerabilities. To apply to our security program please contact **securityteam@arednmesh.org**

CHAPTER

FORTYTWO

FREQUENCIES AND CHANNELS

Example US frequencies and channels that are available for AREDN® networking are shown in the diagram below.

You are responsible for using frequencies, channels, bandwidths, and power levels that comply with your country's amateur radio license requirements.

900 MHz

Channel	4	5	6	7	
Ctr Freq	907	912	917	922	
Status	Shared with US unlicensed				

2.4 GHz

Channel	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8 *
Ctr Freq	2.387	2.392	2.397	2.402	2.407	2.412	2.417	2.422	2.427	2.432	2.437	2.442	2.447
Status	non-US only		*Unshared*		Cannot Use	Shared with US unlicensed							

* Only 5 MHz channel width is available on channel 8

3.4 GHz

Channel	76	77	78	79	80	81	82	83	84	85	86	87	88	89
Ctr Freq	3.380	3.385	3.390	3.395	3.400	3.405	3.410	3.415	3.420	3.425	3.430	3.435	3.440	3.445
Status	US Amateur operations remain on a secondary basis but are subject to removal at any time by FCC notice*													

* per FCC 20-138 IV-E-69

5.8 GHz

Channel	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
Ctr Freq	5.655	5.660	5.665	5.670	5.675	5.680	5.685	5.690	5.695	5.700	5.705	5.710	5.715	5.720	5.725	5.730	5.735	5.740
Status	Shared with US unlicensed indoor/outdoor DFS & Radar Avoidance															Shared with Unlicensed...		

149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	
5.745	5.750	5.755	5.760	5.765	5.770	5.775	5.780	5.785	5.790	5.795	5.800	5.805	5.810	5.815	5.820	5.825	5.830	
Shared with US unlicensed indoor/outdoor																		

167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	
5.835	5.840	5.845	5.850	5.855	5.860	5.865	5.870	5.875	5.880	5.885	5.890	5.895	5.900	5.905	5.910	5.915	5.920	
...Shared with Unlicensed				Shared with US unlicensed mainly indoor									*Shared with Intelligent Transportation System*					

CHAPTER

FORTYTHREE

ACRONYMS LIST

Contributor: Tim Bratton K5RA

List of acronyms sometimes encountered in the AREDN® project.

AAM
AREDN® Alert Message

AAMM
AREDN® Alert Message Manager

AC
Alternating Current

ACK
Acknowledge, Acknowledgement

ADMIN
Administration, Administrative, Administrator

ADS-B
Automatic Dependent Surveillance – Broadcast

ADV
Advanced

AFS
AREDN® Firmware Selector

AGL
Above Ground Level

AM
Amplitude Modulation

AP
Access Point

API
Application Programming Interface

APK
A compressed archive used as the new OpenWRT Package format

APRS
Automatic Packet Reporting System

AREDN
Amateur Radio Emergency Data Network

ARES
Amateur Radio Emergency Service

ARP
Address Resolution Protocol

ARRL
American Radio Relay League

ASF
Apache Software Foundation

ATA
Analog Telephone Adapter

AV
Audio Visual

AVL
Automatic Vehicle Location

AZ
Azimuth

BIN
Binary File Type

BSSID
Basic Service Set Identifier

CAD
Computer Aided Dispatch, Computer Aided Design

CCA
Clear Channel Assessment

CGI
Common Gateway Interface

CHAN
Channel

CIDR
Classless Inter-Domain Routing

CM
Centimeter (10 E-2 meter)

CMD
Command

CONF
Configuration

CPU
Central Processing Unit

CSMA/CA
Carrier Sense Multiple Access / Collision Avoidance

CSS
Cascading Style Sheets (computer graphics)

CSV
Comma Separated Values

CTR
Center

CTRL
Control (a keyboard key)

CTS
Clear to Send

CTCSS
Continuous Tone-Coded Squelch System

DB
Decibel

DBI
Decibel relative to an isotropic radiator

DBM
Decibel relative to one milliwatt (10E-3 watt)

DC
Direct Current

DDNS
Dynamic Domain Name System

DFS
Dynamic Frequency Selection

DHCP
Dynamic Host Control Protocol

DHS
Department of Homeland Security

DMZ
Demilitarized Zone

DNS
Domain Name System

DSP
Digital Signal Processor

DTD
Device to Device

Ed25519
Security key type (Edwards-curve Digital Signature Algorithm)

EL
Elevation

EMCOMM
Emergency Communications

EPUB
Electronic Publication

ETX
Expected Transmission

EULA
End user license agreement

EWMA
Exponential Weighted Moving Average

FCC
Federal Communications Commission

FLV
Flash video file container format

FQDN
Fully Qualified Domain Name

FTP
File Transfer Protocol

GHZ
Gigahertz (10E9 Hertz)

GIS
Geographic Information System

GNU
A project within which the free software concept originated

GPS
Global Positioning System

GUI
Graphical User Interface

GZ
A file format and software application (gzip) for file compression/decompression (g for GNU)

HaLow
Sub-gigahertz WiFi protocol providing long range at low power levels

HLS
HTTP Live Streaming

HSL
Hue, saturation, lightness (for image color representation)

HTML
HyperText Markup Language

HTTP
HyperText Transfer Protocol

IANA
Internet Assigned Numbers Authority

ICMP
Internet Control Message Protocol

ICS
Incident Command System

ICS-213
General Message form

ID
Identifier, Identity

IEEE
Institute of Electrical and Electronic Engineers

IIC
A synchronous master/slave single-ended, serial communications bus

IMAP
Internet Mail Access Protocol

IMG
Image file format

IPK
A compressed archive used as the original OpenWRT package format

IP
Internet Protocol, Internet Protocol address

IRC
Internet Relay Chat

IRCD
Internet Relay Chat Daemon

ISES
Open Information Systems for Emergency Services

ISP
Internet Service Provider

I2C
A synchronous master/slave single-ended, serial communications bus

JSON
Javascript Object Notation

KML
Keyhole Markup Language

KW
Kilowatt (10E3 watt)

LAMP
Linux, Apache, mySQL/MariaDB, and Perl/PHP/Python

LAN
Local Area Network

LAT
Latitude

LED
Light Emitting Diode

LGI
Long Guard Interval

LGPL
Lesser General Public License (GNU)

LON
Longitude

LOS
Line of Sight

LQ
Link Quality

LQM
Link Quality Monitor

LUCI
OpenWRT web interface toolkit

MA
Milliamp (10E-3 amp)

MAC
Media Access Control

MACOS
Apple Computer Operating System

MAMPP
MacOS, Apache, mySQL/MariaDB, and Perl/PHP/Python

MB
Megabtye (10E6 bytes)

MBPS
Megabits (10E6 bits) per second

MCC
Mobile Command and Control

MCS
Modulation Coding Scheme

MG
Magnetic North

MHZ
Megahertz (10E6 Hertz)

µV
Microvolt (10E-6 volt)

MIMO
Multiple Input Multiple Output

MM
Millimeter (10E-3 meter)

MP4
A digital multimedia container format

MRC
Maximal Ratio Combining

MTR
My traceroute, Matt's traceroute – combines functions of traceroute and ping into one network diagnostic tool

MS
Microsoft

MSC
Mobile Switching Center

MSG
Message

NAT
Network Address Translation

NAV
Navigation

NB
Nightly Build

NC
Non-commercial

NLOS
Near Line of Sight

NLQ
Neighbor Link Quality

NTP
Network Time Protocol

NVRAM
Non-volatile Random Access Memory

OFDM
Orthogonal Frequency Division Multiplexing

OLSR
Optimized Link State Routing protocol

OLSRD
Optimized Link State Routing Daemon

ONVIF
Open Network Video Interface Forum

OPENWRT
: An open-source project for embedded operating systems based on Linux upon which AREDN® is based

OPKG
: OpenWRT package management utility

OS
: Operating System

PBX
: Private Branch Exchange

PC
: Personal Computer

PDF
: Portable Document Format

PEP
: Peak Envelope Power

PHP
: Perl Hypertext Pre-Processor

PKT
: Packet(s)

PPK
: Private Key File Extension

POE
: Power Over Ethernet

POP
: Post Office Protocol

PR
: Pull Request (a GitHub mechanism)

PSK
: Phase-Shift Keying

PSK
: Protect Access Pre-Shared Key

PTMP
: Point to multipoint

PTP
: Point to point

PTZ
Pan, Tilt, and Zoom (video camera control)

PUB
Public Key File Extension

PUTTY
Communications tool for running interactive command-line sessions on other computers

PVE
Proxmox Virtual Environment

PVID
Port VLAN Identification

PXE
Preboot Execution Environment

QEMU
Quick Emulator (computer virtualization engine)

QTH
Radio Q-signal for "Location"

RACES
Radio Amateur Civil Emergency Service

RAM
Random Access Memory

REPO
Repository in GitHub

RF
Radio Frequency

ROM
Read-Only Memory

RPI
Raspberry Pi single-board computer

RSA
Security key type (Rivest–Shamir–Adleman)

RSSI
Received Signal Strength Indicator

RST
ReStructured Text file format

RTS
Request to Send

RTSP
Real Time Streaming Protocol

RX
Receive, Receiver

SCP
Secure Copy Program

SDR
Software Defined Radio

SF
San Francisco (California)

SISO
Single Input Single Output

SKYWARN
Program of the National Weather Service which collects reports of localized severe weather in the United States

SMS
Short Message Service

SMTP
Simple Mail Transfer Protocol

SNMP
Simple Network Management Protocol

SNR
Signal to Noise Ratio

SOP
Standard Operating Procedure

SQL
Structured Query Language for relational databases

SS
Spread Spectrum

SSH
Secure Shell

SSID
Service Set Identifier

SSL
Secure Sockets Layer

TCP
Transmission Control Protocol

TCP/IP
Transmission Control Protocol/Internet Protocol

TDMA
Time Division Multiple Access

TELNET
Command line terminal program

TFTP
Trivial File Transfer Protocol

TMP
temporary

TN
True North

TOH
OpenWRT Table of Hardware

TX
Transmit, Transmitter

UCI
OpenWRT Unified Configuration Interface

UDP
User Datagram Protocol

UI
User Interface

URL
Universal Resource Locator

USB
Universal Serial Bus

V
Volts

V2V
Virtual-to-virtual VM migration program

VDC
Volts – Direct Current

VLAN
Virtual Local Area Network

VLC
VideoLAN Client

VM
Virtual Machine

VMDK
Virtual Machine Disk format

VOIP
Voice over IP

W
Watt (unit of power)

WAN
Wide Area Network

WEBRTC
Web Real-Time Communication - open-source project to facility web communications

WG
WireGuard

WGT
Weight

WIFI
Family of wireless networking protocols based on IEEE 802.11 standard

WIMAX
Family of wireless communication protocols based on IEEE 802.16 standard

WIN
Microsoft Windows

WINLINK
Worldwide radio messaging system using amateur radio frequencies

WINSCP
Secure file copy program for Windows

WISP
Wireless Internet Service Provider

WPA
WiFi Protected Access encryption method (WPA/WPA2/WPA3)

WX
Weather

XLINK
Cross-Link configured to pass AREDN® data between non-AREDN® devices

XMPP
Extensible Messaging and Presence Protocol

YAAC
Yet Another ARPS Client

ZIP
File format and software application used for file compression/decompression

CHAPTER

FORTYFOUR

LICENSE

44.1 Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

44.1.1 Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

- **Considerations for licensors:** Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors.
- **Considerations for the public:** By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the

licensor's permission is not necessary for any reason–for example, because of any applicable exception or limitation to copyright–then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public.

44.1.2 Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

44.1.3 Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

c. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

d. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

e. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

f. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

g. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

h. **NonCommercial** means not primarily intended for or directed towards commercial advantage or monetary compensation. For purposes of this Public License, the exchange of the Licensed Material for other material subject to Copyright and Similar Rights by digital file-sharing or similar means is NonCommercial provided there is no payment of monetary compensation in connection with the exchange.

i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

44.1.4 Section 2 – Scope.

a. **License grant.**

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

A. reproduce and Share the Licensed Material, in whole or in part, for NonCommercial purposes only; and

B. produce and reproduce, but not Share, Adapted Material for NonCommercial purposes only.

2. **Exceptions and Limitations.** For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. **Term.** The term of this Public License is specified in Section 6(a).

4. **Media and formats; technical modifications allowed.** The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The

Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. **Downstream recipients.**

A. **Offer from the Licensor – Licensed Material.** Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. **No downstream restrictions.** You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. **No endorsement.** Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. **Other rights.**

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties, including when the Licensed Material is used other than for NonCommercial purposes.

44.1.5 Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. **Attribution.**

1. If You Share the Licensed Material, You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

For the avoidance of doubt, You do not have permission under this Public License to Share Adapted Material.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

44.1.6 Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database for NonCommercial purposes only and provided You do not Share Adapted Material;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

44.1.7 Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

44.1.8 Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

    1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

    2. upon express reinstatement by the Licensor.

    For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

44.1.9 Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

44.1.10 Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.