# AREDN Documentation

*Release 3.22.1.0*

**AREDN**

**May 10, 2022**

# GETTING STARTED GUIDE

Link: AREDN Webpage


**Release** 3.22.1.0

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.

---

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

---

Link: AREDN Webpage

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 between network nodes.

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 might be 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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

TWO

# 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 come with the vendor's firmware pre-installed, but by following a few simple steps 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 (Open Wireless Router) and OLSR (Optimized Link State Routing protocol).

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 in the Supported Platform Matrix.

There is additional guidance on the features and characteristics of specific devices in the Device Selection Chart on the AREDN® website.

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.
- Many devices have an integrated dual-polarity MIMO (Multiple Input-Multiple Output) antenna which helps to leverage multipath propagation. AREDN® has always supported and recommended using MIMO hardware, since these devices typically outperform single chain radios when used as mesh nodes.
- 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, but firmware images continue to grow in size and functionality. 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, 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.

---

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

THREE

# DOWNLOADING AREDN® FIRMWARE

## 3.1 Current Stable Releases

Once you have selected and obtained a device, the next step is to choose the matching AREDN® firmware image for that specific device. The AREDN download page displays the most current firmware releases for every supported device.

Locate your device model/version in the left column. Most manufacturers print the hardware version on the product package label. In some cases, though, you may need to start the device using the manufacturer's pre-installed firmware and navigate to the system information page to determine the hardware version.

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

- If you are loading AREDN® firmware on a device for the first time you must download the *factory* firmware from the middle column. For Mikrotik devices you must also download the *sysupgrade* image from the righthand column.
- If you are already running AREDN® firmware on the node then you will choose the *sysupgrade* firmware from the righthand column, and you will use the AREDN® web interface to perform the firmware upgrade.

Once you have selected the correct firmware image for your device, click the link to download the image file to your local computer. Make a note of the download location on your computer, since you will need to use that image to install the AREDN® firmware on your device.

**Features Inherited from OpenWRT for New Architectures** The latest AREDN® firmware contains features which are inherited from the newest OpenWRT upstream releases. The OpenWRT *Release Notes* describe these new features and can be found here: OpenWRT 19.07 Release Notes and OpenWRT 21.02 Release Notes.

One important change is the inclusion of a new *target* (architecture) for the firmware, labelled "ath79", which is the successor to the existing "ar71xx" target. The OpenWRT team explains the new target here: ath79. Their main goal is to bring the code into a form that will allow

all devices to run a standard unpatched Linux kernel. This will greatly reduce the amount of customization required and will streamline the firmware development process.

Since not all supported devices have been migrated to the new "ath79" target, AREDN® continues to build firmware for both targets. **You should select the latest recommended target image based on the type of hardware on which it will be installed.** Refer to the latest firmware notes in order to ensure you have the correct firmware image for your specific device.

## 3.2 Nightly Build Firmware

*Nightly Build* firmware contains the latest bug fixes, features, and support for the newest devices being added to the *Supported Platform Matrix*. It is considered more experimental or cutting-edge and may not be suitable for production nodes. However, 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 has just been added, then it might make sense to install the nightly build firmware.

To download the *Nightly Build*, navigate to the Software > Nightly Builds link on the AREDN® website. Nightly Build filenames are prefixed with *aredn-XXXX-yyyyyyy*, where *XXXX* identifies the build number and *yyyyyyy* is a unique software commit identifier. You will find the most recent *README* file, a cumulative list of the changes included in the build, and a link to download the firmware. As explained above, select the correct target architecture for the device you will be flashing. To return your device to the current stable release, download the correct *Stable Release* firmware and reflash your device.

Be aware that when a new nightly build becomes available, any older builds automatically become obsolete. If you want to install add-on packages for nodes running a nightly build, understand that specific packages will not be available for an *older* build if a *newer* build has superseded it. Be sure to upgrade to the current nightly build before installing packages. Nightly build firmware contains the cumulative changes that have gone before, so review the *Changelog* to determine which features are included in the build.

Link: AREDN Webpage

Link: AREDN Webpage

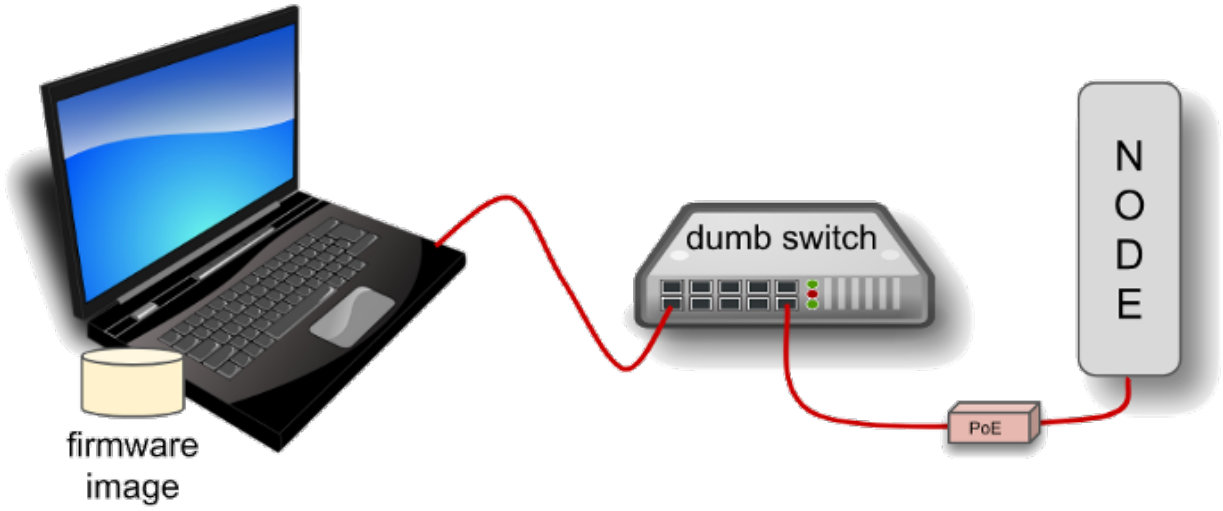
CHAPTER

FOUR

# INSTALLING AREDN® FIRMWARE

There are two cases for installing AREDN® firmware:

1. If you already have an existing version of AREDN® running on your device, then you can use your computer's web browser and navigate to **Setup > Administration > Firmware Update** to install your new firmware. This process will be explained in more detail in the **Configuration Deep Dive** section of this guide. Also, see *Firmware Upgrade Tips* in the **How-to Guides** section for additional information.
2. If you are installing AREDN® firmware on a device for the first time, each hardware platform may require a unique procedure.




The diagram above shows that your computer with the downloaded firmware image must be connected to the node using Ethernet cables in order to install the AREDN® image. It is helpful to connect the computer and node through a simple Ethernet switch so that the switch can maintain the computer's network link even when the node is rebooting.

Different node hardware will require different methods for installing the AREDN® firmware. For **Ubiquiti** devices, your computer's TFTP client will connect to the node's TFTP server in order to upload the firmware image. For **Mikrotik** and **TP-LINK** devices, your computer will run a PXE

server and the node's remote boot client will download the boot image from your computer. For **GL-iNet** devices, your computer's web browser will connect to the node's web server to upload the firmware image. Refer to the specific procedures below for your node hardware.

If you experience an issue uploading firmware to your device you can refer to the *Firmware Tips* document in the **How-To Guide**.

## 4.1 Firmware First Install Checklists

It may be helpful to have a brief checklist of steps to follow when doing the initial firmware installation on node hardware. The checklists below are provided to assist with this process, based on the manufacturer of your device. Complete step-by-step instructions are detailed in the sections that follow.

`GL.iNet First Install Checklist (PDF)`

`Mikrotik First Install Checklist (PDF)`

`TP-LINK First Install Checklist (PDF)`

`Ubiquiti First Install Checklist (PDF)`

## 4.2 Ubiquiti First Install Process

**Ubiquiti** devices have a built-in TFTP server to which you can upload the AREDN® *factory* image. Your computer must have TFTP client software available. Linux and Mac both have native TFTP clients, but you may need to enable or obtain a TFTP client for Windows computers. If you are using a Windows computer, enable the TFTP client or download and install another standalone TFTP client of your choice.

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 192.168.1.100.

   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 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]
> put <full path to the firmware file>
  [For example, put /temp/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
```

   The TFTP client should indicate that data is being transferred and eventually completes.

5. Watch the LEDs for about 2-3 minutes until the node has finished rebooting. The reboot is completed when the LED 4 light (farthest on the right) is lit and is steady green.

6. Configure your computer's Ethernet network interface to use DHCP for obtaining an IP address from the node. You may need to unplug/reconnect the Ethernet cable from your computer to force it to get a new IP address from the node.

7. After the node reboots, open a web browser and use either `http://192.168.1.1` or `http://localnode.local.mesh` for the URL. 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.

8. Click the *Setup* button and configure the new "firstboot" node as described in the **Basic Radio Setup** section.

## 4.3 Mikrotik First Install Process

Mikrotik devices require a **two-part install** process: First, boot the correct mikrotik-vmlinux-initramfs file with the **elf** extension, and then use that temporary AREDN® Administration environment to complete the installation of the appropriate *sysupgrade* file with the **bin** extension.

Mikrotik devices have a built-in PXE client which allows them to download a boot image from an external source. Your computer must run a **PXE Server** to provide an IP address and boot image to Mikrotik devices. The important functions of a **PXE** server are to give the node an IP address via DHCP as well as providing the firmware image via TFTP. The reason AREDN® suggests using the 192.168.1.x network on your **PXE** server is to eliminate the need 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.

**Preparation**

- Download *both* of the appropriate Mikrotik *factory* and *sysupgrade* files from the AREDN® website. Rename the **elf** 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 because it will put devices on the network you will eventually need to use in order to complete the installation. For example, you can give your computer a static IP such as 192.168.1.100 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, 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. If you are flashing a *Mikrotik hAP ac lite* that uses a separate AC adapter, connect the last Ethernet cable from the switch to the Mikrotik's WAN port (1).

**PXE Boot: *Linux Procedure***

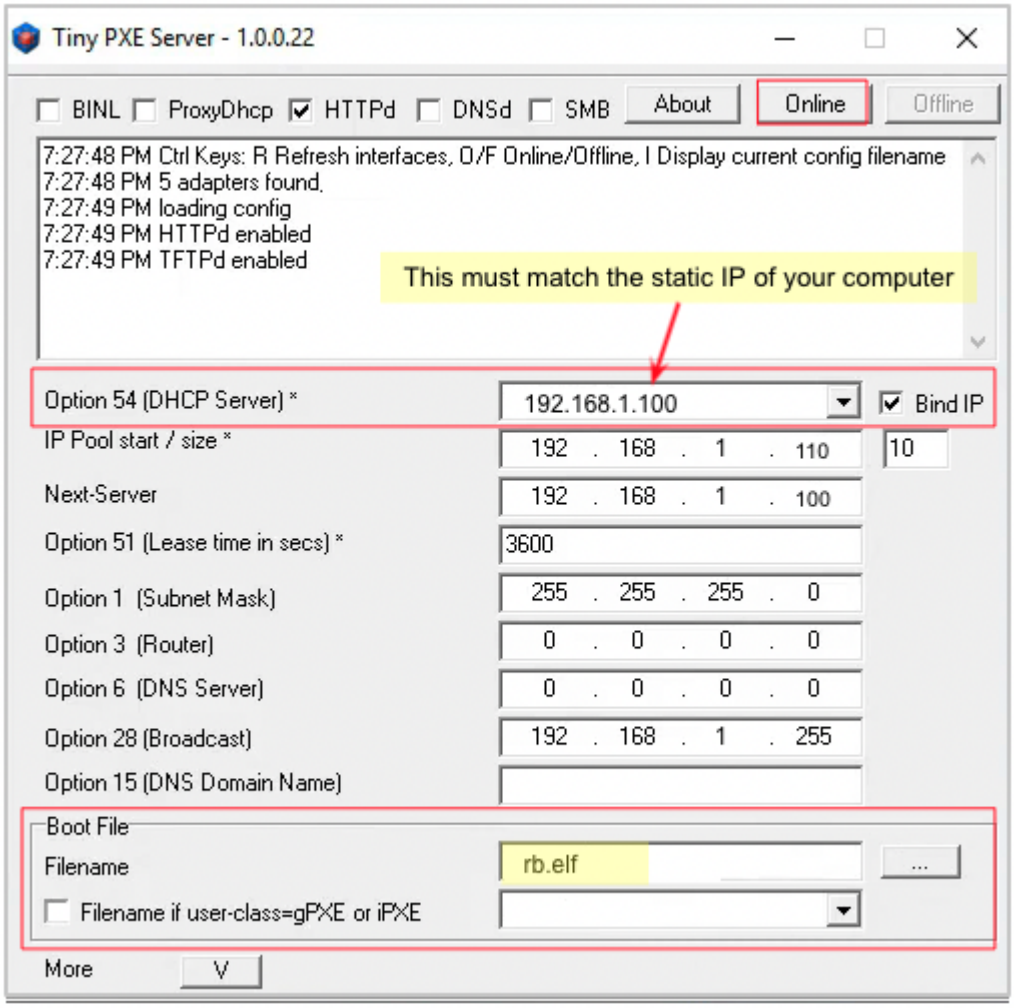
1. Create a directory on your computer called `/tftp` and copy the `rb.elf` file there.
2. Determine your computer's Ethernet *interface name* with `ifconfig`. It will be the interface you set to 192.168.1.100 above. You will use this interface name in the command below as the name after `-i` and you 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 192.168.1.110,192.168.1.120 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=192.168.1.110,192.168.1.120 -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.

**PXE Boot:** ***Windows Procedure*** You will need to install and configure a PXE Server on your Windows computer. The example below uses *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.

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 click the *Offline* button in *Tiny PXE*. You are finished using *Tiny PXE* when the **elf** image has been read by the node.

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

### Install the *sysupgrade* Firmware Image

1. After booting the **elf** 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.

> **Attention:** For the *Mikrotik hAP ac lite* **only**, 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.

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.

2. In a web browser, open the node's Administration page `http://192.168.1.1/cgi-bin/admin` (user = 'root', password = 'hsmm') and immediately navigate to the *Firmware Update* section. Browse to find the *sysupgrade* **bin** file you previously downloaded and click the *Upload* button.

   As an alternative to using the node's web interface, you can manually copy the *sysupgrade* **bin** file to the node and run a command line program to install the firmware. This will allow you to see any error messages that may not appear when using the web interface. Note that devices running AREDN® firmware images use port 2222 for secure copy/shell access.

   Execute the following commands from a Linux computer:

   ```
   >>>
   my-computer:$ scp -P 2222 <aredn-firmware-filename>.bin␣
    ↪root@192.168.1.1:/tmp
   my-computer:$ ssh -p 2222 root@192.168.1.1
   ~~~~~~~ after logging into the node with ssh ~~~~~~~
   node:# sysupgrade -n /tmp/<aredn-firmware-filename>.bin
   ```

   To transfer the image from a Windows computer you can use a *Secure Copy* program such as WinSCP. Then use a terminal program such as PuTTY to connect to the node via ssh or telnet in order to run the sysupgrade command shown as the last line above.

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

## 4.4 TP-LINK First Install Process

**TP-LINK** devices may or may not allow you to use the manufacturer's pre-installed *PharOS* web browser 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 do not need to follow any 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 have a built-in PXE client which allows them to obtain new firmware from an external source. Your computer must run a **PXE Server** to provide an IP address and boot image to the device. The important functions of a **PXE** server are to give the node an IP address via DHCP as well as providing the firmware image via TFTP. The reason AREDN® suggests using the 192.168.1.x network on your **PXE** server is to eliminate the need 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.

**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 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 because it will put devices on the network you will eventually need to use to complete the installation. For example, you can give your computer a static IP such as 192.168.1.100 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, 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 192.168.1.100 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 192.168.1.110,192.168.1.120 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=192.168.1.110,192.168.1.120 -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.

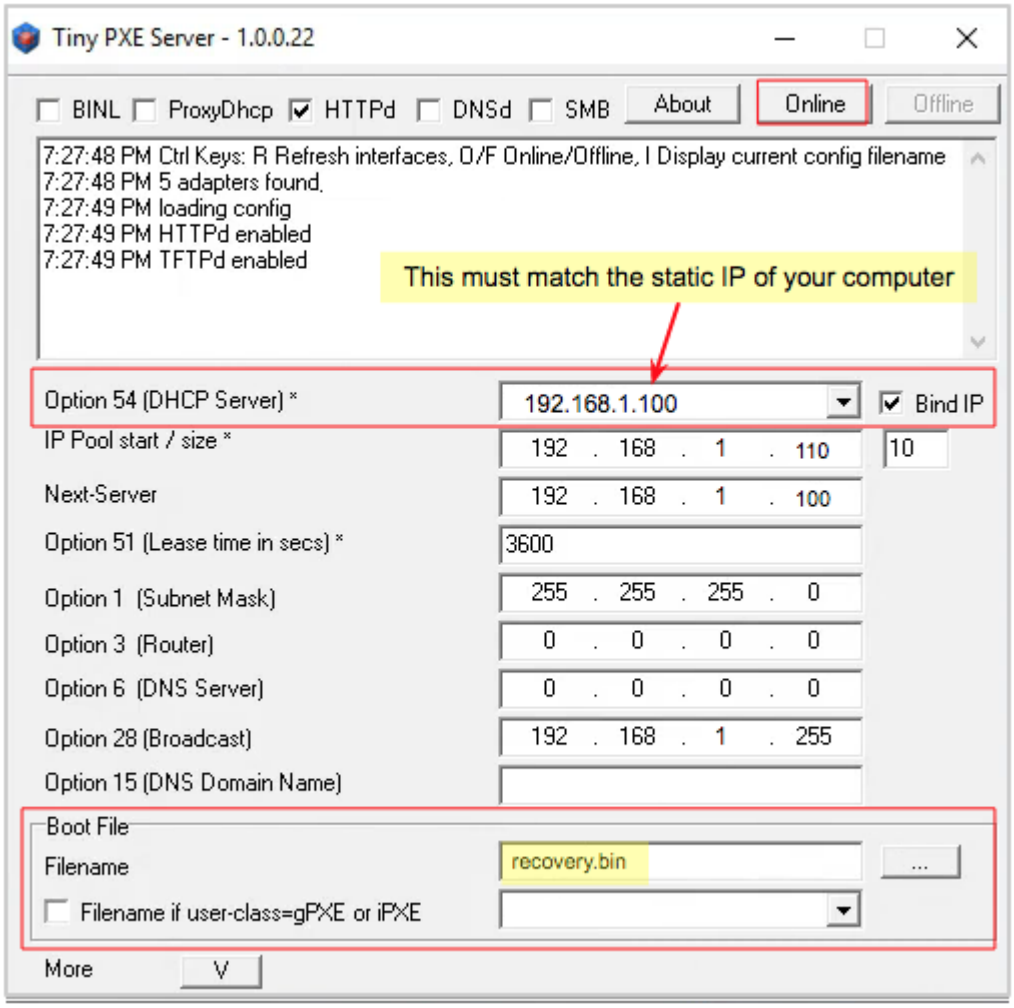
5. The node will now automatically reboot with the new AREDN® firmware image.

**Windows Procedure** You will need to install and configure a PXE Server on your Windows computer. The example below uses *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.

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 click the *Offline* button in *Tiny PXE*. You are finished using *Tiny PXE* when the firmware image has been read by the node.
6. The node will now automatically reboot with the new AREDN® firmware image.

## 4.5 GL-iNet First Install Process

**GL-iNet** 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. Both GL-iNet and AREDN® 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 downloaded for your device.

> **Attention:** Be sure to uncheck the **Keep Settings** checkbox, since GL.iNet settings are incompatible with AREDN® firmware.

The node will automatically reboot with the new AREDN® firmware image. If for some reason your GL-iNet device gets into an unusable state, you should be able to recover using the process documented here: GL-iNet debrick procedure

## 4.6 After the Firmware Install

After the node reboots, it should have a default IP address of 192.168.1.1. By default AREDN® devices provide DHCP (Dynamic Host Control Protocol) on their LAN interface, so your computer will receive an IP address automatically from the node. Ensure that your computer is set to obtain its IP address via DHCP.

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.

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`. 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 **Basic Radio Setup** section of the documentation.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

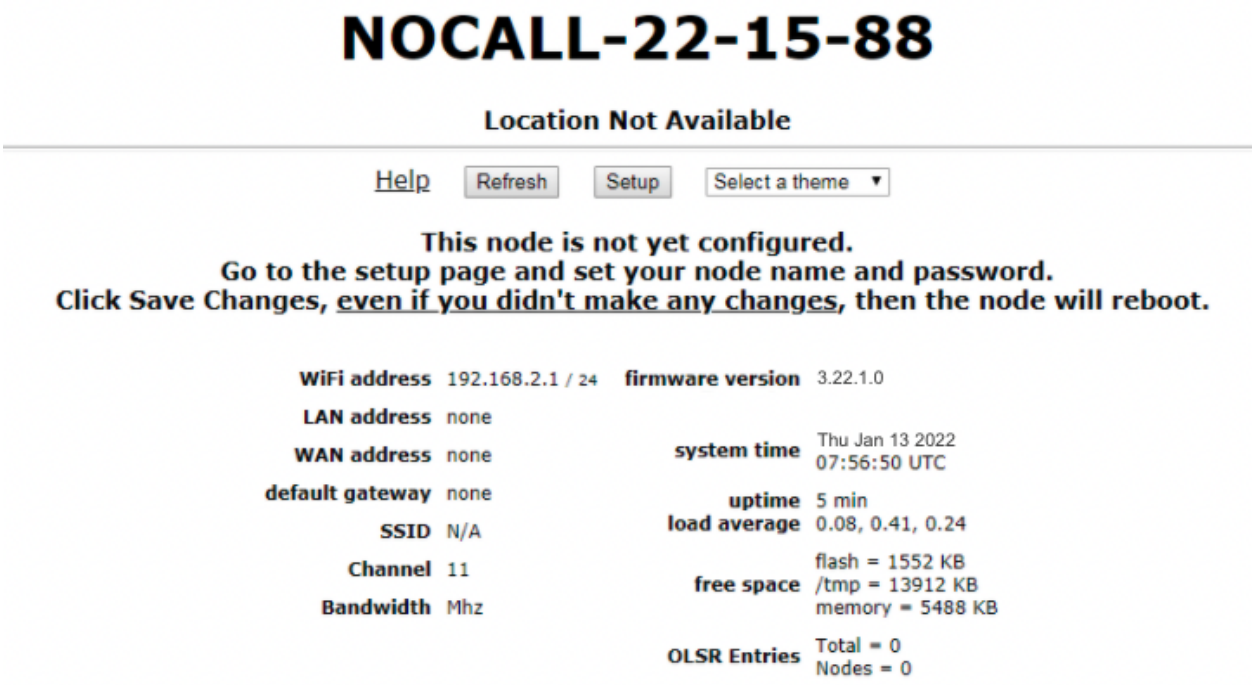
FIVE

# BASIC RADIO SETUP

## 5.1 First-Time Setup

After you have installed the AREDN® firmware, rebooted the device, and connected your computer to a LAN port on the node or the PoE unit, you can navigate to your node's web interface at `http://192.168.1.1` or `http://localnode.local.mesh` for first-time radio setup. 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. Each node will serve its web interface on ports 80 and 8080.

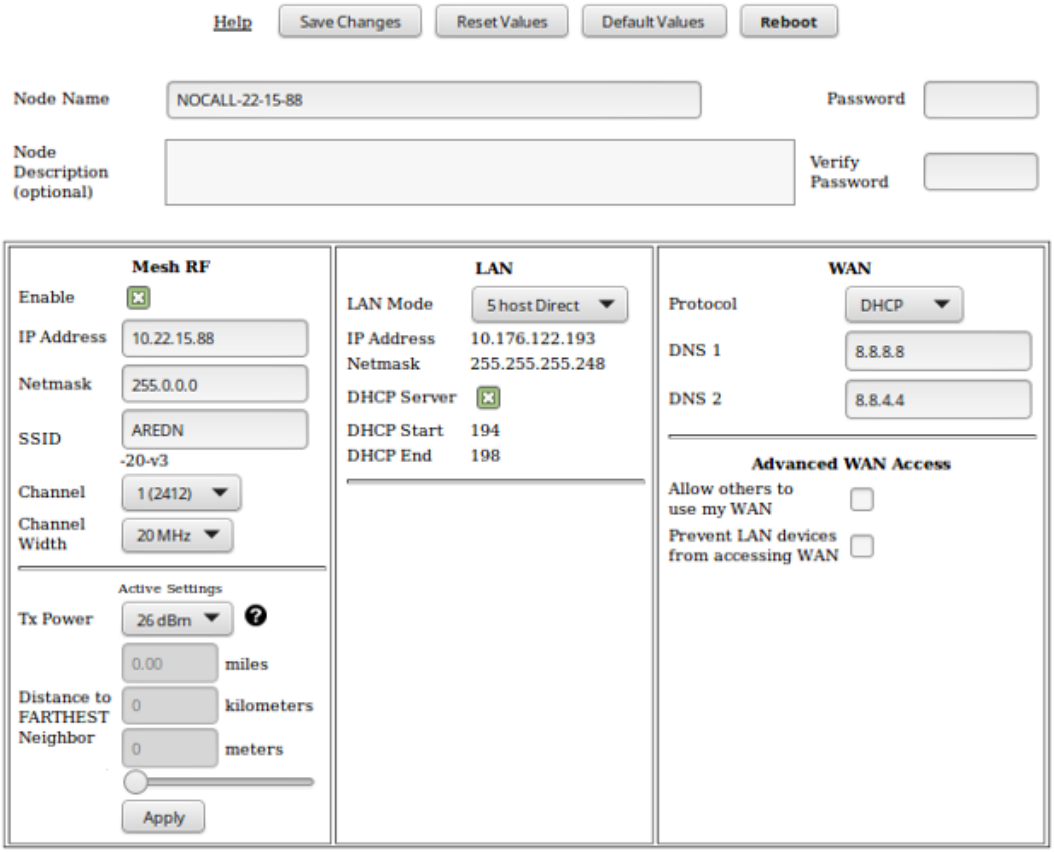
The initial status page will be displayed, instructing you to configure your node by clicking the **Setup** button. This is sometimes referred to as the "NOCALL" or *firstboot* display.



You will be prompted to enter the administrative login credentials. The default authentication credentials are:

Username: `root`
Password: `hsmm`

The **Basic Setup** page will be displayed, as shown below.




Many of these settings will be described in detail in subsequent sections of this documentation. In order to get your new AREDN® node on the air for the first time, you need to enter the following items.

**Node Name** Begin the node name with your callsign, followed by 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. Node names are not case sensitive, but the case will be preserved on the node status display. Amateur radio operators are required to identify all transmitting stations. The AREDN®

node name is beaconed automatically by the node every five minutes, so the node name must contain your callsign. 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. Enter it again in the *Retype Password* box to verify it is correct. The first time a node is configured it will require you to change the password. Be sure to remember or record the new password so you can use it for any future administrative tasks on the node.

**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, the radio model and antenna specifications, or the tactical purpose for the node. There are no character restrictions in the field, but the maximum length allowed is 210 characters.

**Mesh RF** The *IP Address, Netmask, and SSID* fields are automatically calculated for you based on the unique MAC (Media Access Control) address of your node. Do not change these settings. Everything under the **LAN** and **WAN** columns can be left unchanged for now.

**Channel and Channel Width** Nodes communicate only with other nodes that use the same channel and channel width. You can determine the correct settings by talking with other local node operators to find out which settings are required for joining their networks.

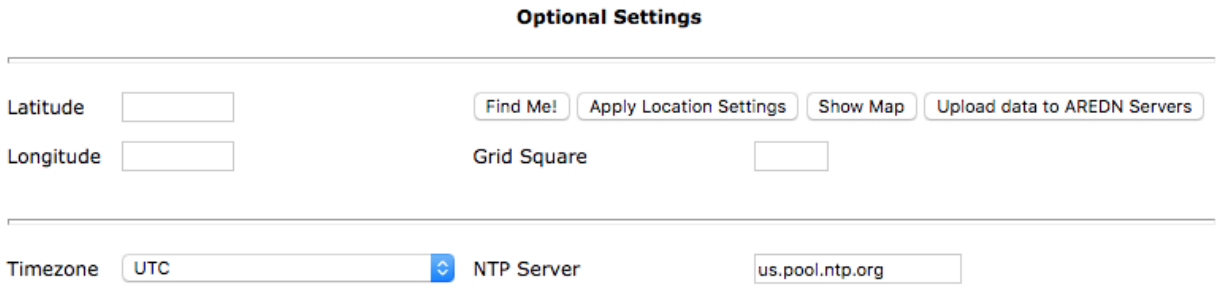
**Active Settings** See the *Configuration Deep Dive* section for more information about these and other settings in the **Mesh RF** column.

- Use the dropdown list to select the maximum output power for this device. Remember that amateur operators are required to use the minimum power necessary to make contact with other stations.
- Use the slider to select the maximum distance you estimate between your node and other neighboring nodes. The default value is *zero* which tells the node to automatically determine the correct distance value. See the *Configuration Deep Dive* section for additional information.
- Some devices have max power levels that change depending on the channel or frequency being used, and in that case the max level may change when you save the settings. The output power will be capped at the max level supported by the hardware for that frequency.
- Once these settings have been adjusted, click the **Apply** button.

Once you have entered, applied, and verified that your node settings are correct, click the **Save Changes** button. Your node will record the new configuration settings and automatically reboot.

## 5.2 Optional Settings

In this section you can enter your node's latitude and longitude, as well as the grid square designator. The latitude/longitude values should be in decimal format (for example, 30.5432 and -95.1234). The optional node location settings are not required in order for your node to function normally.




- If you are using a location-aware web browser, you can click the **Find Me** button to populate the latitude/longitude fields. This works well if you are viewing the *Basic Settings* page on a mobile device with built-in GPS.
- If your node has an Internet connection available, the **Show Map** and **Upload Data to AREDN Servers** buttons will become active. The **Show Map** button will display a map that allows you to click the position where your node is located or to drag an existing location marker to a different spot on the map. Both of these actions will automatically update the latitude/longitude fields on the page.
- The **Upload Data to AREDN Servers** button will send your node information to an AREDN® server on the Internet. By submitting this information you are agreeing to allow AREDN® to publish your node location on a public mapping service and utilize the information for other purposes such as statistical analysis. No sensitive data such as passwords are sent to the AREDN® servers. If you wish to remove your node location from the public mapping service, simply clear or erase your latitude/longitude values, click *Apply Location Settings* and then *Upload Data to AREDN Servers*.
- Click the **Apply Location Settings** button after entering new location information on this page. The new settings become active without clicking the *Save Changes* button.

You may also change the timezone for your node's system time, as well as selecting a Network Time Protocol (NTP) source if your node is connected to a network which has a network time server.

Link: AREDN Webpage

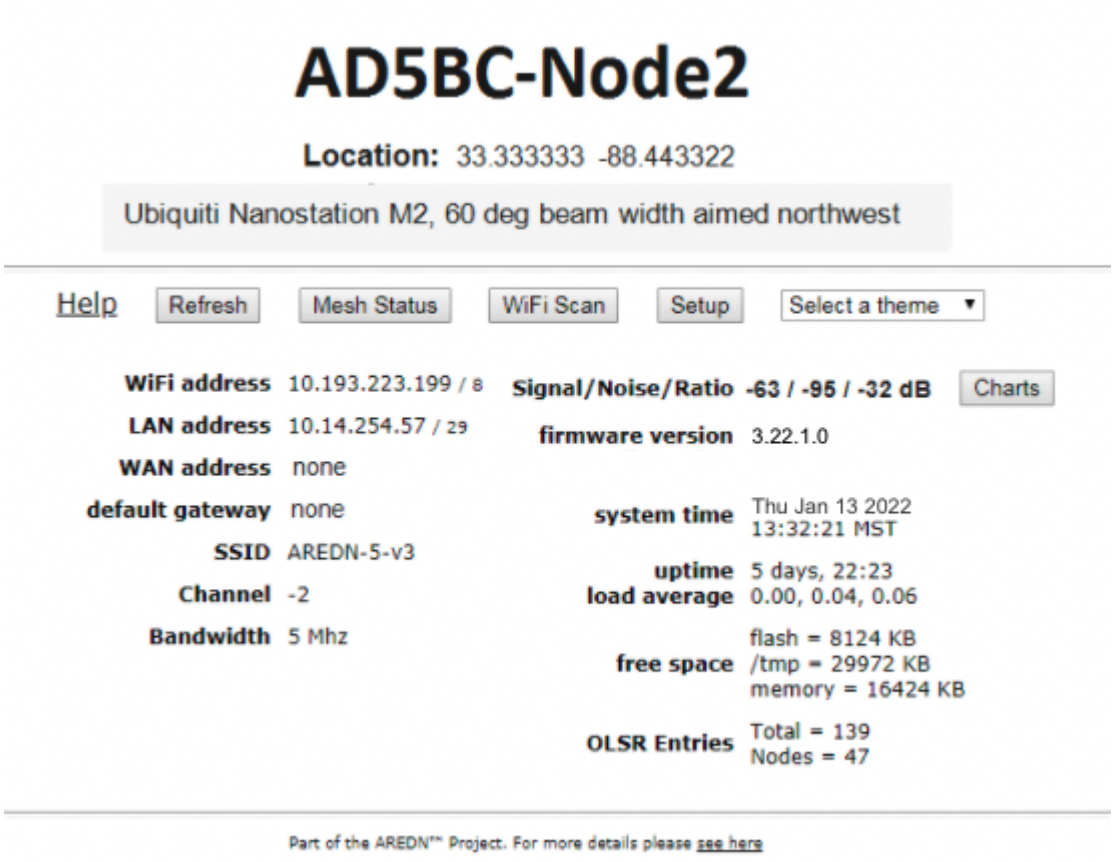
Link: AREDN Webpage

CHAPTER

SIX

# NODE STATUS DISPLAY

Once you have completed the initial setup on your AREDN® node, you can connect your computer to the LAN port on the PoE and navigate to the following URL: `http://localnode`. You will be redirected to the **Node Status** page as shown below.




Below the node name bar there are several controls.

**Help** Opens a new window or tab to display the node help page.

**Refresh** Updates the Node Status page with current data.

**Mesh Status** Opens the **Mesh Status** page showing the neighbor nodes and remote nodes visible on the mesh network, as well as what services are being provided by those nodes.

**WiFi Scan** Displays a list of other 802.11 signals that your node can see. The 802.11 signals may include Access Points, neighbor nodes, and other mesh networks (foreign ad-hoc networks). WiFi Scan only finds devices on the same channel width as your node is configured to. When installing at a new location, it is best practice to scan on 5, 10, and 20MHz channels to find all 802.11 signals in range. This information will help to pick a channel clear of other interference. When multiple ad-hoc networks are visible (with different SSIDs or channels), the ID of each 802.11 ad-hoc *network* is displayed but not the individual nodes. There is also an automatic scan mode, but running a Wifi Scan continuously is not recommended, particularly if the node is actively routing traffic. The scan is passive, or only listens for other beacons through all channels, and risks loss of data on the assigned channel. Wifi Scan does not transmit probes on every channel in passive mode, thus no risk of interfering with Radar stations on DFS channels, or other unintended transmissions. Multiple attempts of Wifi Scan will be necessary to find all devices in range.

**Setup** Navigates to the **Setup** pages for your node. You will need to supply a username and password to access those pages. The username is always `root`, while the password is the one you set during initial node setup. If the node has not yet been configured, the password is `hsmm`.

**Select Theme** AREDN® firmware has several built-in display themes. The default `aredn` theme has a gray background with black and red text. The `black_on_white` theme is often chosen because it provides the best screen contrast on a computer exposed to direct sunlight. `red_on_black` is much better suited for nighttime use since it helps preserve night vision.

## 6.1 Node Settings Summary

The area under the display controls shows both configuration and network status information. The left column contains the IP address details for the network interfaces on this node, as well as the SSID, channel, and bandwidth settings.

The right column contains the Signal Strength readings and other attributes of your node. The **Signal/Noise/Ratio** shows the strongest neighbor radio signal strength in dBm (decibels relative to one milliwatt) from all connected stations, and it is available only when the node is connected by RF (Radio Frequency) to a mesh network. Click these links for further information about Signal to Noise Ratio and values measured in decibels.

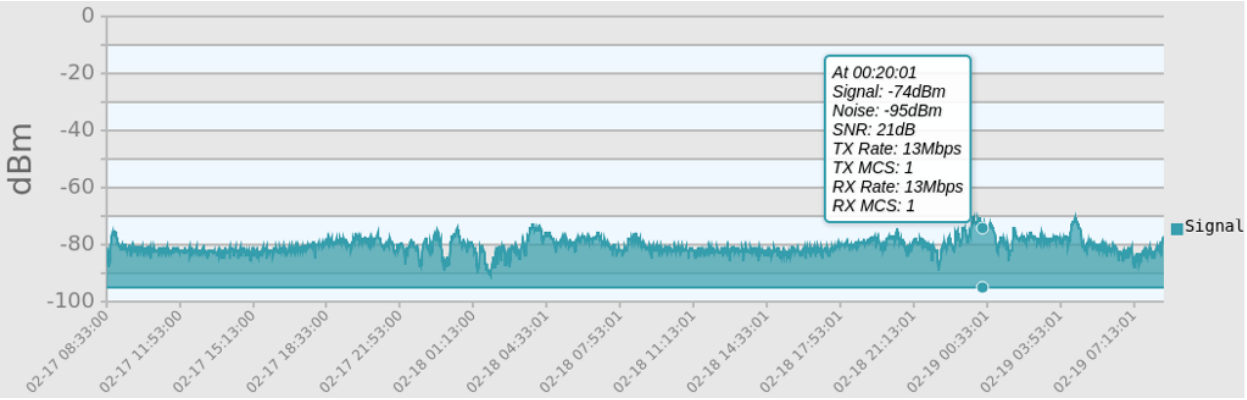
Below the Signal Strength readings are the node's **Firmware Version** and network type. The **System Time** is displayed, as well as the **Uptime**, or time since the last reboot. Nodes have no internal battery or realtime clock, so the time is reset every time the node is booted. If an Internet connection becomes available, the internal NTP (Network Time Protocol) client will connect with a time server to sync the node's time.

The **Load Average** is the average number of processes that have been running on the node for the last 1, 5, and 15 minutes. **Free Space** tells you how much space is available on local storage devices. Flash is the internal non-volatile storage where the operating system, configuration files, and software packages are kept. /tmp is a filesystem in memory that stores the node's current status and various temporary files. **Memory** is the amount of RAM (Random Access Memory) available for running processes on the node.

The OLSR (Optimized Link State Routing protocol) **Entries** show the total number of entries in the routing table, as well as the number of nodes currently connected to the mesh network.

## 6.2 Signal Charts

There is a **Charts** button next to the node's **Signal Strength** display, and clicking this button takes you to **Signal Charts**. This page shows RF signal information in both a realtime and an archived view. The default view shows the average signal of all connected stations in realtime.




At the top of the charts display there are several control buttons.

**Archive** This button shows the charts for any archived signal data on this node. Statistics are stored on the node in a circular buffer which holds about two days of data.

**Realtime** This button shows the charts for current signal data as seen from this node.

**Quit** This button exits the charts view and takes you back to the *Node Status* page.

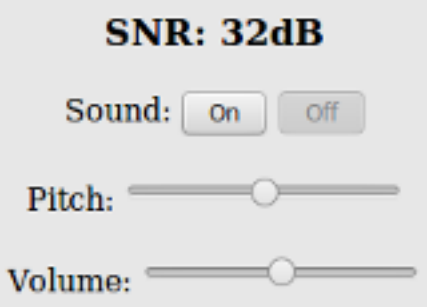
Below these controls you can choose to view the signal strength statistics for individual nodes that are directly connected to your node. Choose the neighbor node from the **Selected Device** dropdown list. Changing the selected device will automatically reload the chart to show that node's information.

Hovering over data points within a chart will show additional information for each data point, including Time, Signal, Noise, SNR (Signal to Noise Ratio), TX Rate, TX MCS (Modulation Coding Scheme), RX Rate, and RX MCS. If no traffic is being routed to the neighbor, the Rate and MCS values may be zero until data is available. An MCS value of zero may indicate non-802.11n encoding schemes (ie. 802.11a/b/g).

The small icon with three vertical dots in the upper right corner of the chart allows you to download a snapshot of the chart to a graphic file on your local computer (jpeg or png).

Data shown in the **Archive** charts is not stored in permanent memory on the node. The node will store approximately two days of archived data, and all data is cleared when a node is rebooted.

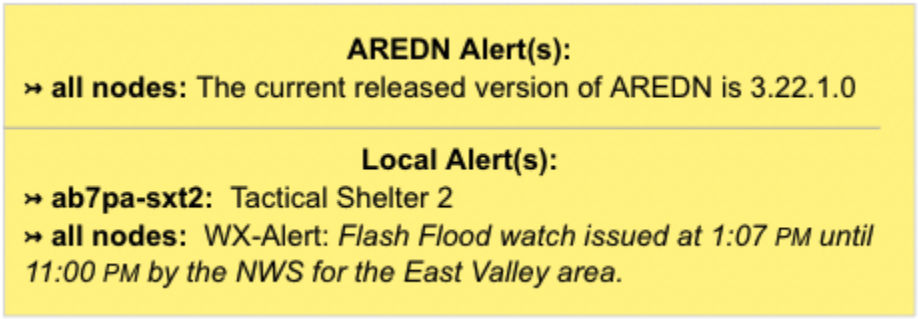
If you click and drag your mouse across a region of the chart, the display will zoom into that selected area. This allows you to view data points for a specific time range of your choice. While zoomed, two additional icons will appear in the upper right of the chart. The **Pan** icon allows you to scroll and pan the zoomed portion of the chart. The **Reset** icon returns the chart to its normal display mode.




On the left of the Realtime Graph there is an **SNR Sound** control. Clicking the *On* button will cause your computer to emit a tone that corresponds to the relative SNR level, with higher pitch tones indicating better SNR. This feature was added in order to provide an audio queue to operators in the process of aligning directional antennas. When your antenna reaches a position at which the highest pitch tone is heard you can lock it down without having to look at the signal graph display, knowing that you are receiving the best signal available. You can also adjust the tone pitch and volume with the sliders on the sound control.

## 6.3 AREDN® Alert Messages

The AREDN® development team has the ability to post messages which Internet-connected nodes will automatically retrieve once every 12 hours by default. There are two types of messages: broadcast messages intended for all nodes, and directed messages which are only retrieved by individual nodes. Messages are displayed in a yellow banner on a node's webpages above the node name. Be aware that there is no guarantee of privacy for these messages, since anyone can view the message repository online.




Mesh nodes without Internet access also have the ability to display *Local Alerts*. The process for setting up a local message repository is described in the **Configuration Deep Dive** section. If a node has Internet access as well as local messages, then both types of messages will be displayed in the AREDN® alerts banner as shown in the example above.

Link: AREDN Webpage

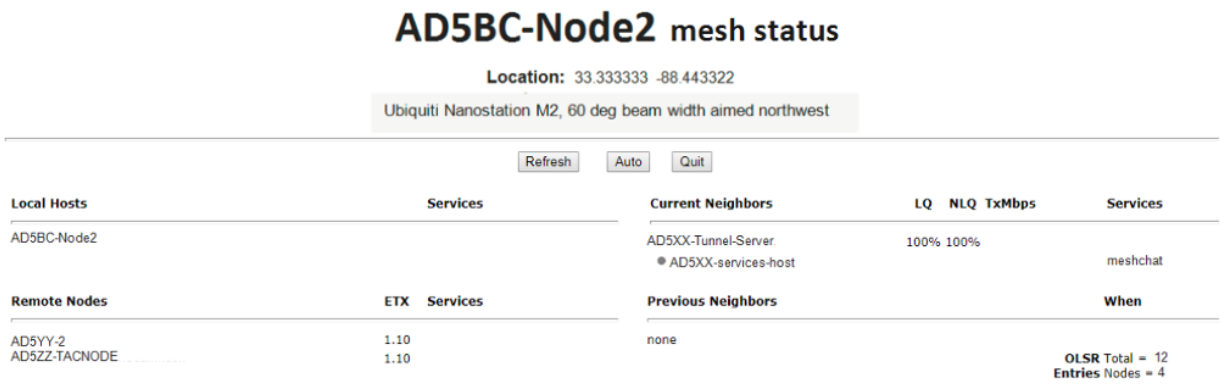
Link: AREDN Webpage

CHAPTER

SEVEN

# MESH STATUS DISPLAY

The **Mesh Status** page lists mesh nodes, link quality information, and the advertised services on the mesh network.




Below the node name bar there are several controls.

**Refresh** This button refreshes the **Mesh Status** display with current information.

**Auto** This button sets the display to automatically refresh the node information every 10 seconds. To end auto-refresh mode, click **Stop** or **Quit**. **Stop** returns to the static *Mesh Status* display. **Quit** takes you back to the *Node Status* display, and clicking *Mesh Status* again from there will return you to auto-refresh mode on the *Mesh Status* display.

**Quit** This button returns you to the *Node Status* display.

There are four sections on the **Mesh Status** display.

**Local Hosts** This shows your mesh node along with any connected hosts and the advertised services available on your node and hosts. Typically you may click the service name to open a new browser tab containing the features of that service. This will be true for any available services in the *Current Neighbors* or *Remote Nodes* sections.

If you have any hosts for which you selected *Do Not Propagate* in the **DHCP Reservations List**, those hosts will be displayed in a light gray color only on that node's *Local Hosts* column. If you created any **DNS Aliases** for your hosts, those aliases will be displayed in a light orange color only on that node's *Local Hosts* column. All other hosts will be displayed in the default color for the theme that you are using.

**Current Neighbors** This shows a list of *Neighbor Nodes* that are directly connected with your node (1 hop). These nodes may be connected via RF, DTD (Device to Device) link using an Ethernet cable, or a tunnel over an Internet connection. There are several link quality statistics displayed for each connected node.

- `LQ` or Link Quality is your node's view of the percent of OLSR (Optimized Link State Routing protocol) packets received from the neighbor node. These packets exchange mesh routing and advertised services information, and they include a sequence number that is used to identify missing packets which is a measure of the quality of the link.
- `NLQ` or Neighbor Link Quality is the neighbor node's view of the percent of OLSR packets received from your node. This measures the quality of the link from the neighbor's side.
- `TxMbps` or Transmit Megabits per Second is a calculated estimate of the data rate achieved across the link with the neighbor node. This column may show zero if the data being transmitted between these nodes is not sufficient for the metric to be calculated.
- `Services` is the column where any available services on the neighbor node will be displayed. You may click on the service link to navigate to the webpage for that service on the neighbor node.

In addition to the neighbor node name, there may be a text abbreviation in parentheses that tells how the neighbor node is connected.

- `(dtd)` indicates a *Device to Device* connection using an Ethernet cable between the nodes. The neighbor may be listed twice if both an RF and DTD path exist.
- `(tun)` indicates the path to the neighbor node is over an Internet tunnel. `(tun*?)` next to a mesh node in the *Remote Nodes* column indicates the node has tunnel links over the Internet to connect mesh islands together. `?` is a number indicating the number of tunnel connections on that node.
- `(wan)` indicates the node has been configured as a *Mesh Gateway*. Typically this is a gateway to the Internet, but it may also be to another isolated network.

**Remote Nodes** This section lists other nodes on the network that are two or more hops away. Advertised services on nodes and their attached hosts are also listed. Remote Nodes are sorted by their `ETX` or *Expected Transmission* metric. ETX (Expected TX metric) is a calculated estimate of the number of OLSR packets that must be sent in order to receive a round trip acknowledgement, and it is often referred to as "link cost". When sending data the OLSR protocol selects the least cost route based on the lowest ETX path in the direction of the final destination.

**Previous Nodes** This section lists any nodes which were recently connected to your node but are not currently connected. It shows the node name or IP address, as well as how long it has been since a node was actively connected to your node.

Link: AREDN Webpage

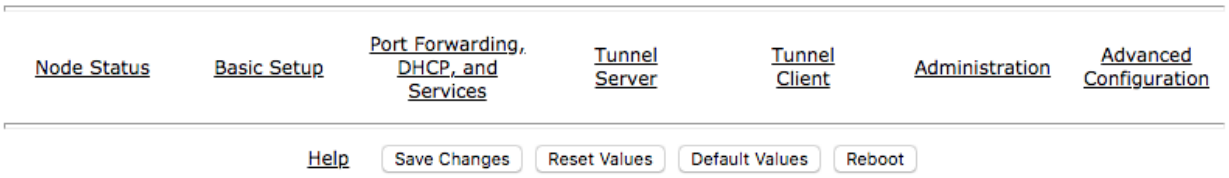
Link: AREDN Webpage

CHAPTER

# EIGHT

# CONFIGURATION DEEP DIVE

During your node's *Basic Setup* you used the configuration display by clicking the **Setup** button and typing your username and password. The configuration area has many additional features which will be described in more detail below. Clicking **Node Status** exits configuration mode without saving any changes, returning you to the *Node Status* display.




There are several control buttons below the configuration links section.

**Help** Opens a new window or tab to display the node help page.

**Save Changes** Click this button to save any configuration changes you have made. Saving changes will first do a basic validation of the new settings, saving them to flash memory if no errors are found. The new settings take effect in about 20 seconds and a reboot may or may not be required.

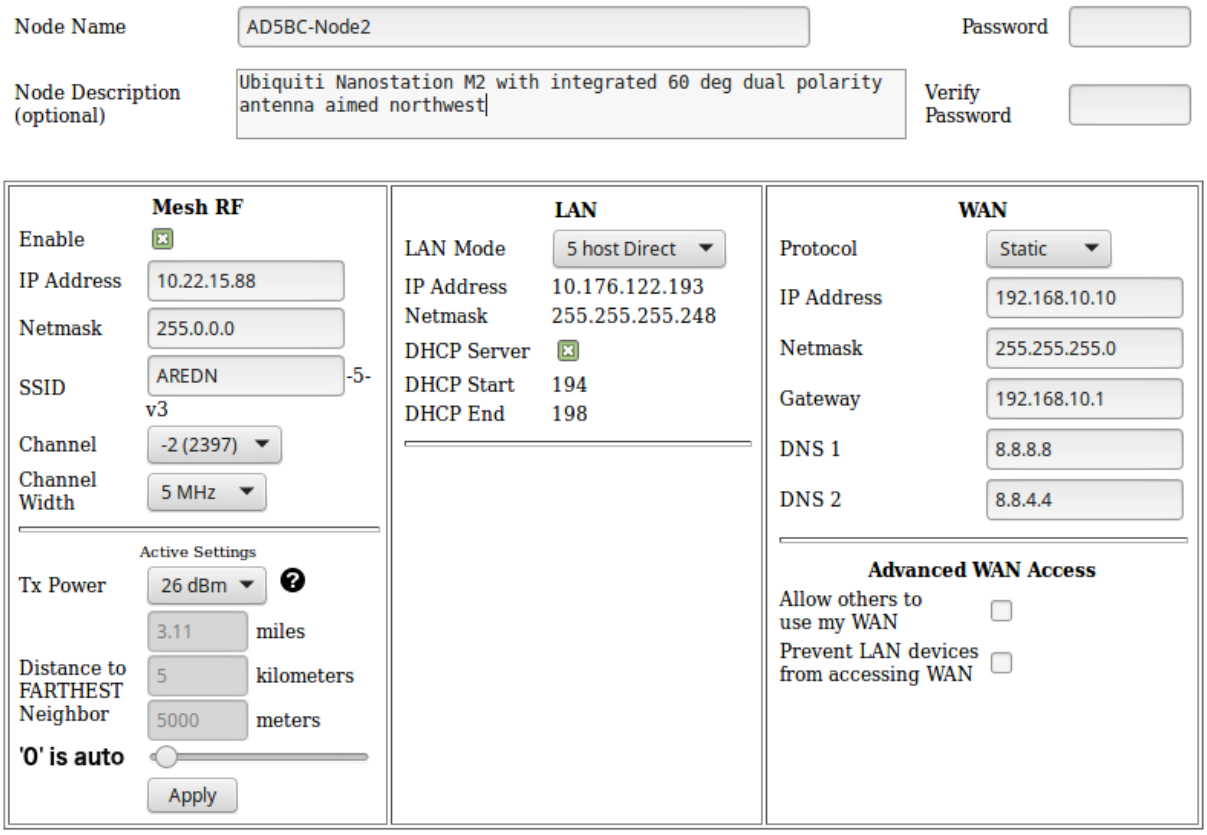
**Reset Values** Click this button to reload the currently saved settings from flash memory, effectively undoing any changes that were made.

**Default Values** Click this button to reset your node's basic settings to the default values. This action does not affect your existing node name.

**Reboot** Click this button to force your node to reboot.

## 8.1 Basic Setup

You have already configured many of the basic settings, but there are several additional features that will be explained below.




### 8.1.1 Mesh RF Column

*Mesh RF* is the node's *radio* interface. The AREDN® firmware has been designed to simplify the process of configuring networking interfaces. Network values are automatically calculated based on the unique MAC addresses of your node. You may need to change the *Channel* and possibly the *Channel Width* parameters to match those of your local AREDN® mesh, as explained previously in the **Basic Radio Setup** section. Normally you will not need to change the other network settings on this page, so keep these values unless you fully understand how the mesh works and why the defaults may not be suitable for your situation.

The **Active Settings** can be adjusted and applied without saving changes or rebooting your node. However, they will return to their original values after a reboot unless you click *Save Changes*. A

node may decrease its output power as it increases its data rate in order to maintain a linear spectrum.

**Distance Setting** The *Distance* setting is only applicable to nodes that can communicate directly over RF. This setting adjusts the RF retry timer to define how long the transmitter will wait for an acknowledgement from a neighbor station. If the distance parameter is too short, the transmitter will send duplicate data packets before an acknowledgement has time to be received. If the distance parameter is too long, the transmitter will wait extra time before considering the data lost and retransmitting the packets.

The maximum distance settings the ath9k wireless driver allows depends on the channel width:

| Channel Width | Maximum Distance |
|---|---|
| 20 MHz | 46666 meters |
| 10 MHz | 103030 meters |
| 5 MHz | 215757 meters |

**Auto-Distance**: A value of zero will cause the radio to automatically determine the RF retry timer by measuring the actual time it takes acknowledgement packets to be received. The timer is set using an Exponential Weighted Moving Average (EWMA). The auto-distance setting is best used on high quality, long distance point-to-point links between backbone or relay nodes. Fifty percent performance increases have been observed on those links compared to using a static distance setting.

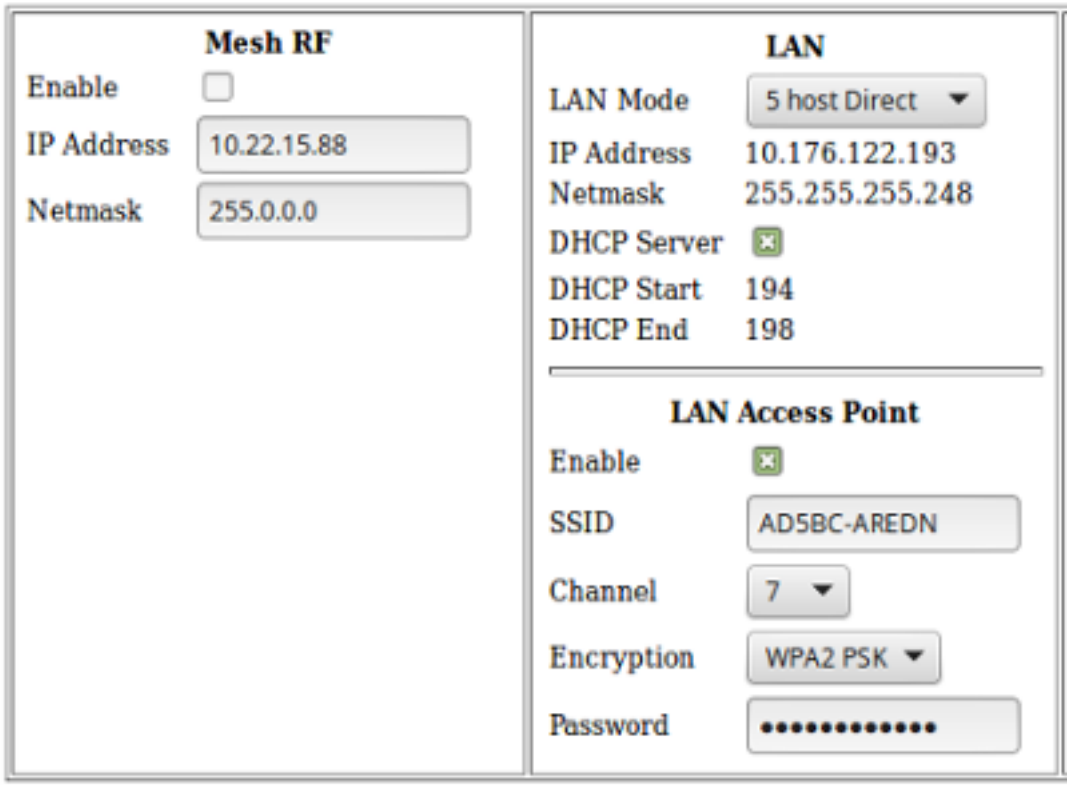
Since auto-distance causes the node to calculate the best value based on actual data flow, it will require both time and adequate data traffic to arrive at the optimal setting. The node may not be able to arrive at the optimal values if a link is not being used to send a significant amount of data, because it starts at the max value and then drops down to the optimal setting. Over time the auto-distance setting should stabilize around the best value.

> **Attention:** The auto-distance setting does **not** work well when nodes are in close proximity, when link quality is marginal, or when there are many nodes sharing the channel. In these cases the round-trip packet timing has a very wide range of values, so the timeout value becomes inflated and inconsistent. Static settings should be used in these situations.

A basic rule of thumb is when nodes are within five kilometers of each other you should test several *static* distance settings to see which one works best. The best way to test each distance setting is to use the **iperf3** package between endpoint nodes to measure the throughput of the RF channel under different distance settings. See *Test Network Links with iperf3* in the **How-To Section** for additional information.

**Enable/Disable Mesh Radio** You can disable your node's radio interface by deselecting the *Enable* checkbox, saving your changes, and rebooting the node. With the Mesh RF interface disabled the *Active Settings* no longer apply and will disappear. Since your node now has an unused RF interface, you will notice that a new section appears which allows you to use the

node's radio as an FCC Part 15 *LAN Access Point*. You can enable or disable the LAN AP using the *Enable* checkbox. See the details below for configuring the LAN Access Point.




### 8.1.2 LAN Column

The LAN column contains the settings for the Local Area Network hosted by the AREDN® node. There are several options under the *LAN Mode* dropdown.

The default mode is `5 Host Direct`. In this mode every host on the LAN has direct access to and from the mesh. This mode was created to reduce the amount of manual configuration 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. With *Direct* mode the LAN shares the same address space as the mesh at large. Port forwarding is not needed because NAT is not used, and there is no firewall between the LAN and the mesh.

The mesh address space is automatically managed, so you cannot configure the LAN network settings in *Direct* mode. The only configurable option available in *Direct* mode is the size of the LAN subnet which can accommodate either 1, 5, 13, or 29 LAN hosts. A one host subnet can be used 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.

It is important not to use a subnet larger than is 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 RF* interface. If a conflict does occur it can be fixed by changing the *Mesh RF* IP address.

The other LAN Mode 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 RF* IP address of the node. This is the same way that most routers use an Internet connection, and all services provided by computers on the LAN can only be accessed through port forwarding rules. A single DMZ (DeMilitarized Zone) server can be used to accept all incoming traffic that is not already handled by other rules or by the node itself.

By default each node runs a DHCP server for its LAN interface, which lets the node assign IP addresses automatically for devices connected to the node's local area network. The last octet of the start/end range for host IP addresses is shown in the LAN column. If you choose to disable the DHCP server, you must manually configure the host IP addresses to be within the LAN network range. There should be only one DHCP server for each IP address scope or range, so you may need to disable your node's DHCP server if there is already another device providing DHCP services on your node's local area network. Click this link for additional information on Dynamic Host Control Protocol.

When you connect a device to your node's LAN, not only should it have an IP address in the LAN IP address range, but it is best practice for LAN devices to obtain their 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 device is moved to another mesh node or network.

If you enabled the *LAN Access Point* feature mentioned previously, edit the access point's SSID, channel, encryption method, and password. Click *Save Changes* to write your information to the node's configuration, and a node reboot will also be required. Now wireless devices can connect to your node through this new WiFi AP, and their DHCP IP address will be assigned by the node's DHCP server. If your node hardware has two radios, for example the *Mikrotik hAP ac lite* with both 2.4 and 5.8 GHz radios in a single unit, the *LAN Access Point* section will always be visible whether or not your *Mesh RF* interface is enabled.

### 8.1.3 WAN Column

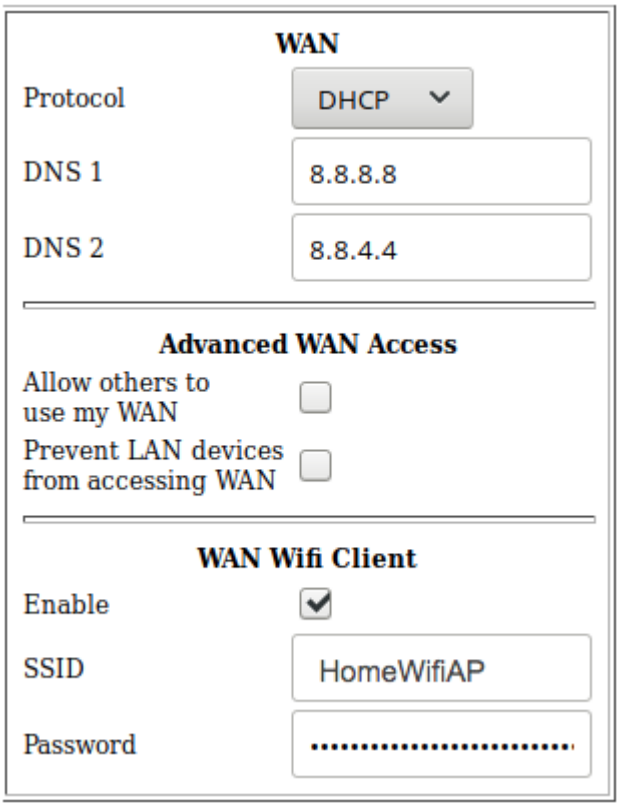
The WAN (Wide Area Network) interface on your node is typically used to connect it to the Internet or to another external network. By default the WAN interface is set to obtain an IP address via DHCP from your upstream network. The DNS (Domain Name System) servers are set by default to use Google's DNS services and should not be changed under normal circumstances. Google's name resolution servers are configured properly to detect error conditions and report them correctly.

If you are not going to use the WAN interface on your node, you can select *disabled* from the *Protocol* dropdown list. If you will be using your node as a *Tunnel Server*, you should reserve an IP

address on your router for the node's WAN interface. This will be explained in the *Tunnel Server* section below.

When a node has Internet access on its WAN interface, that access is available to the node itself and to any computers connected via the LAN port. Checking the *Allow others to use my WAN* box will allow this node to route traffic from *all* its interfaces to/from the Internet or other external network. This box is unchecked by default because it is not desirable to route Internet traffic over the radio 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 wireless Internet access, consider using an FCC Part 15 access point instead of the node's WAN gateway.

The *Prevent LAN devices from accessing WAN* checkbox will tell the node not to advertise that it can be used as a default gateway. This means that computers on the LAN network will lose their route to the Internet or other networks via your mesh node. This checkbox is deselected by default. If this checkbox is selected your LAN hosts will have no access to the Internet even if your node has Internet access on its WAN interface. You may need to disable the default route if your node needs to be connected to two networks at once, such as being wired to the mesh and connected to a local served agency WiFi network.




As mentioned above in the *Mesh RF* section, if your node has a radio on which you have *disabled* Mesh RF and you are not using it as a LAN AP, you can enable this available radio as a WAN interface by checking the **WAN Wifi Client** checkbox. Enter the SSID and authentication string for the wifi AP that you want to connect through for Internet access.

The mesh node uses "WPA2 PSK" encryption to connect to the wifi AP. The password length must be a minimum of 8 and maximum of 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.

After you have saved changes and rebooted, the node will have Internet access via wifi rather than requiring a cable plugged into the node's WAN port. In fact, enabling the *WAN Wifi Client* will disable VLAN1, so Internet access will no longer be possible through the physical WAN port.

### 8.1.4 Node VLANs

Many of the devices used as AREDN® nodes have only one Ethernet port, but more than one type 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.

**VLAN 1** Packets received by the node that are tagged for VLAN 1 will be identified as WAN traffic from the Internet or another external network.

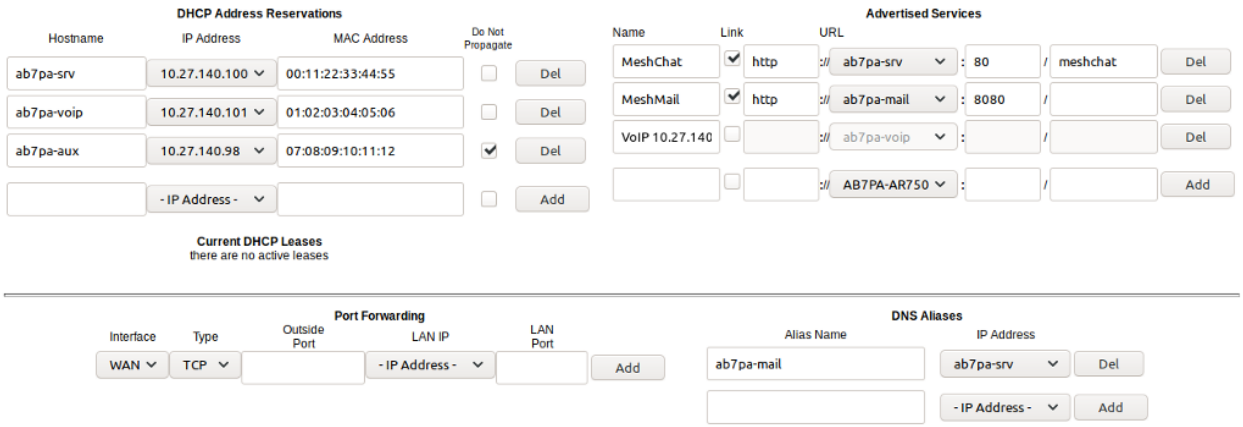
**VLAN 2** Packets received by the node that are tagged for VLAN 2 will be identified as traffic from a DtD node directly connected via Ethernet cable.

**No VLAN tag** Packets received by the node that are untagged will be identified as LAN traffic from computers on the local area network.

It is important to understand AREDN® VLANs when configuring network smart switches for Internet access, 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. Also, on the AREDN® GitHub site there is more information about node VLANs that have been preconfigured in the firmware images for specific types of radio hardware. For additional information visit this link: Ethernet Port Usage

## 8.2 Port Forwarding, DHCP, and Services

Click the **Port Forwarding, DHCP, and Services** link to navigate to these settings. This section provides a way for you to configure LAN network address reservations and service advertisements on your node. If your LAN network uses `NAT` mode, you may also need to define port forwarding rules.




If your node is running its default DHCP server on the LAN network, it will automatically provide IP addresses to connected hosts. Look under the **Current DHCP Leases** heading to see the existing hosts and their assigned IP address.

> **Attention:** The hostnames of computers connected to the mesh at large must be unique. Typically you should prefix your amateur radio callsign to the computer's hostname in order to have the best chance of it being unique on the mesh network.

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. You can permanently reserve that host's DHCP address by clicking the *Add* button at the right of the row in the *DHCP Leases* list. You will see that host now appears in the list under the **DHCP Address Reservations** heading above the list of leases.

There may be some devices on which you are not able to set the hostname prefixed by your callsign. Once you add that device to your **DHCP Address Reservations**, however, click the *Hostname* box to edit the hostname what will be propagated across the mesh network. You may also want to assign a specific IP Address to the device by selecting it from the drop-down list. If you have a device which needs to be reachable on its host node, but which should not be accessed across the mesh network, click the *Do Not Propagate* checkbox to prevent OLSR from propagating that information to the mesh.

### 8.2.1 Advertised Services

*Services* include the required applications, programs, or functions that are available to devices on the mesh network. 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.

Remember that AREDN® nodes have a limited amount of system resources with which to run services, so installing add-on services directly on the mesh node should be avoided because the node will become unstable and the mesh network can fail if insufficient RAM is available for the node to function, particularly on devices with only 32 MB of memory. It is a best practice to run services on an external computer connected to the node's LAN network. In the example above you can see that an external host has been given a reserved DHCP address, and it is also running the *meshchat* program as a service that is advertised on the network through this node. Use the following steps to create an advertised service.

**Name** Enter a service name in the *Name* field.

**Link** Check this box if your want your advertised service to display an active link in the web browser. This allows mesh users to navigate to your service by clicking the link.

**Protocol** Enter the protocol to use in the field between *Link* and *URL*. Common protocols include `http` for website services and `ftp` for file transfer services. Other services may use other protocols.

**URL** From the dropdown list select the node or host on which this service is running.

**Port** Enter the network port on which the service is listening for user 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 (as in the example above). In other cases the network port alone uniquely identifies the application or program that is listening for user connections to that service. You can click this link for additional information about network ports.

Once you have entered the values for your advertised service, click *Add* to add the service to the **Advertised Services** list. You may also remove an existing advertised service by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your changes to the node's configuration.

### 8.2.2 Port Forwarding

If you are using `NAT` for your LAN mode, then *Port Forwarding* rules are the only way other devices have for connecting to your services. To create a port forwarding rule, select the network **Interface** on which the traffic will enter your node. Select the Protocol **Type** used by the incoming packets (TCP, UDP, or Both). Enter the **Port** number that the external request is using to connect to your service. When your node receives traffic on the selected interface, protocol, and port, that request will be routed to the **LAN IP** address and **LAN Port** on which the service host is listening for incoming requests.

Once you have entered these values, click *Add* to add the rule to the **Port Forwarding** list. You may also remove an existing rule by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your port forwarding changes to the node's configuration.

See your node's **Help** file for additional insights on how this configuration section changes based on the LAN mode of your node. Click this link for more information on Port Forwarding.

### 8.2.3 DNS Aliases

**DNS Aliases** provide a way for you to create a mesh alias or synonym for a services computer. This can be useful if you want a computer or device on your node's LAN network to be identified by something other than its actual hostname.

To create an alias, enter an **Alias Name**. 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 drop-down 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, click *Add* to add the alias to the **DNS Aliases** list. You may also remove an existing alias by clicking the *Del* button to delete it from the list. Click the **Save Changes** button to write your changes to the node's configuration.

**Aliases in Direct Mode** When your node is using `Direct Mode` for its LAN, *DNS Aliases* allow your computer or device to be reachable by its alias from across the mesh network. This provides functionality similar to DNS *CNAME* records, so the computer will respond to network requests using its real hostname as well as any aliases that are defined for it.

Once they are defined the **DNS Aliases** become available for creating *Advertised Services* by choosing the alias from the host drop-down selector. This feature can be used for virtual domain email servers, virtual machine identifiers, virtual web site URLs, and many other services.

**Aliases in NAT Mode** *DNS Aliases* work differently in `NAT Mode`. Aliases **cannot** be propagated across the mesh when using `NAT Mode`. They are only visible within the local LAN network on the node. `NAT Mode` aliases **cannot** be used when defining an *Advertised Services* listing. They can only be used as an alternate hostname for a computer or device on the nodes' LAN.

## 8.3 Tunnel Server

Click the **Tunnel Server** link to navigate to these settings. This section provides a way for you to configure your node with unencrypted node-to-node connections across the Internet. Unless you have a specific need for this type of network connection, it is recommended that you do not activate tunnels. This is because it will cause your node to dedicate limited resources to maintaining the tunnel connections. In order to increase the performance of your node you should conserve system resources so they will be available for normal node operations, which is especially important for nodes with limited memory and storage.

Tunnels should be used as a temporary means of connecting mesh islands when RF links have yet to be established. They should be removed as soon as RF links are operational. Remember that AREDN® is first and foremost an emergency communication resource, so it's likely that Internet-dependent links and the assets they provide will not be available during a disaster. Their presence could create a false expectation for served agency personnel, with the possibility that AREDN® might fail to meet their expectations when tunneled resources become unavailable during a disaster.

Also, 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 via RF, then adding one or more tunnel connections on that node will cause the nodes and hosts on the far side of the tunnel(s) to appear on your local *Mesh Status* display. This adds complexity and makes everyone's display a little more difficult to navigate. If you want to participate in remote mesh networks via tunnel, consider establishing those tunnels from one of your nodes that is *not* connected to your local mesh network via RF.

### 8.3.1 Internet Connectivity Requirements

In order to run your node as either a *Tunnel Server* or *Tunnel Client*, you will need to configure additional settings and equipment.

**Managed Switch Settings (both Client and Server networks)** 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 connection on its WAN interface. See the equipment manual for your managed switch to determine how to configure these settings.

---

**Note:** If you are using a *Mikrotik hAP ac lite* or *GL.iNET AR150/AR300M/AR750*, then you do not need a separate VLAN-capable switch as described above. These nodes have built-in switches with the appropriate VLANs preconfigured in the AREDN® firmware.

---

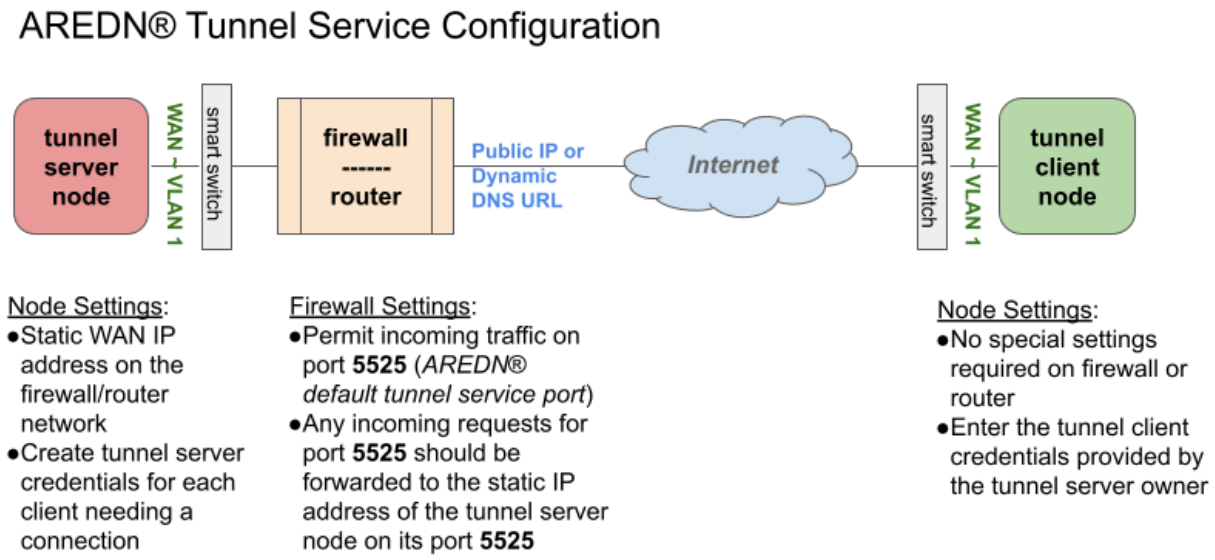
**WAN Interface IP (Tunnel Server *node* only)** Set a static IP address on your tunnel server node's WAN interface so your Internet-connected router/firewall has a consistent way to forward traffic to your node.

**Internet Firewall/Router Settings (Tunnel Server network only)** Set your network firewall or router to permit traffic from the Internet on port 5525, which is the default AREDN® tunnel service port. Then configure a port forwarding rule on your firewall or router to send any traffic from the Internet on port 5525 to the static IP address of your node's WAN interface on the *node's* port 5525. See the equipment manual for your firewall or router to determine how to configure these settings. Also, some Internet Service Providers may not allow port forwarding by default, so you should check with your ISP if you have difficulty opening ports.

Also, remember that the tunneling feature on AREDN® nodes was not compiled with Secure Sockets Layer (SSL) libraries and that tunnel traffic is unencrypted.

### 8.3.2 Tunnel Server Node Settings

The following diagram shows an overview of tunnel services between two nodes.




The tunnel network address ranges are automatically calculated, and it is not necessary to change these settings unless there is a specific reason why the defaults will not work for your situation.

**Tunnel Server DNS Name** Enter the *Public IP Address* or the *Dynamic DNS URL* by which Internet-connected nodes can reach your network.

**Client Node Name** Enter the exact node name of the client node that will be allowed to connect to your node over the tunnel. Do not include the "local.mesh" suffix.

**Client Password** Enter a complex password that the client node will use to connect to your node over the tunnel. Use only uppercase and lowercase characters and numbers in your password.

**Contact Info/Comment (optional)** You have the option to enter a line of text which may include the contact information of the person responsible for the tunnel endpoint. It is a 50 character freeform text field which can contain any other useful identifier or information as needed.

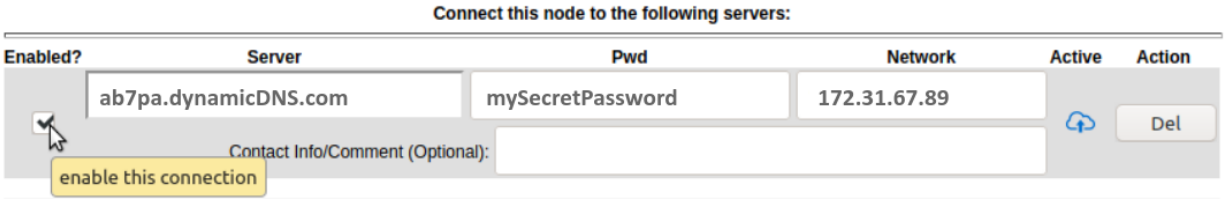
Once these settings are correct, click *Add* to add the new client to the list of authorized tunnel clients. On the right of each entry there is an envelope icon which will automatically open your computer's email program and copy the client settings into a new email which allows you to quickly and easily send credentials to the owners of the client nodes.

To allow a client to connect to your tunnel server, select the **Enabled?** checkbox and click the **Save Changes** button. When a tunnel connection becomes active, the cloud icon at the right of each row will change to indicate that the tunnel is active. Depending on the timing of the webpage refresh, you may need to press the **Refresh** button to see the active icon.

## 8.4 Tunnel Client

Click the **Tunnel Client** link to navigate to these settings. In this section you can configure your node to connect over the Internet to another node running as a *Tunnel Server*. You should already have your VLAN-capable network switch configured as explained in the *Tunnel Server* section above.

Contact the amateur operator who controls the tunnel server and request client credentials by providing your specific node name. The tunnel server administrator will provide you with the public IP or DDNS (Dynamic Domain Name Service) URL for the tunnel server, the password you are to use, and the network IP address for your client node. Enter these values into the appropriate fields on your node and click *Add* to create a client entry in the list.

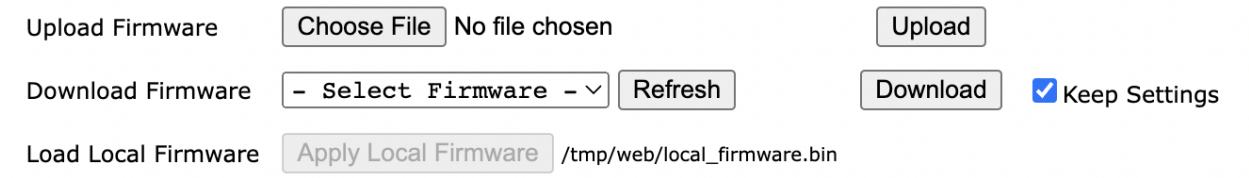



To allow your client to connect to the tunnel server, select the **Enabled?** checkbox and click the **Save Changes** button. When a tunnel connection becomes active, the cloud icon at the right of each row will change to indicate that the tunnel is active. Depending on the timing of the webpage refresh, you may need to press the **Refresh** button to see the active icon.

## 8.5 Administration

Click the **Administration** link to navigate to these settings. There are four sections that provide ways for you to manage the firmware, packages, security keys, and support data on your node.

**Firmware Update** There are currently three ways to update the firmware on your node. No matter which method you choose, you can retain your existing configuration settings by selecting the *Keep Settings* checkbox.

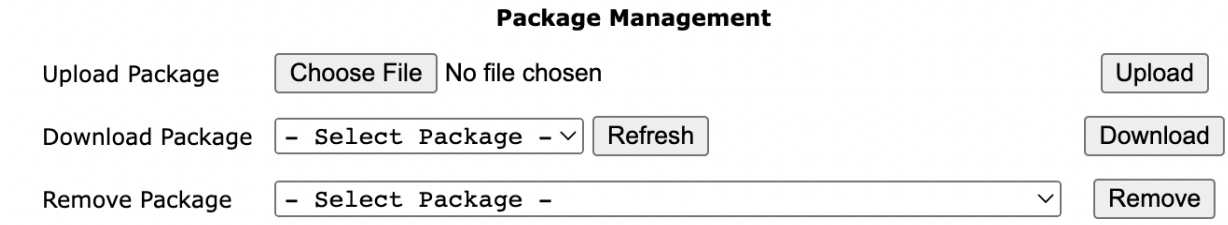



1) **Upload Firmware**: If you have a new firmware image that you have already downloaded to your computer from the AREDN® website, click the *Browse* button and select the firmware file from the location on your computer where you saved it. Click *Upload* and the file will be uploaded and installed on the node.

2) **Download Firmware**: If your node has Internet access you can use the *Download Firmware* option. Click *Refresh* to update the list of available images. The source URLs that are queried are those listed on the *Advanced Configuration* page of your node. Select the image to download, click *Download*, and wait for the firmware to download and be installed.

3) **Load Local Firmware**: If you need to 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/web` folder, and name it `local_firmware.bin`. Refresh your node's *Administration* page and once the page detects the `/tmp/web/local_firmware.bin` file, then the *Apply Local Firmware* button will become active. Press this button to begin the update process using the firmware you previously uploaded.

**Package Management** Here you can install or remove software packages on the node. *Upload Package* allows you to install a package file by uploading it from your computer to your node. *Download Package* allows Internet-connected nodes to retrieve a package from the AREDN® website. Clicking *Refresh* will update the list of packages available for download.




The *Remove Package* list shows all packages currently installed on the node. Selecting a package and clicking *Remove* will uninstall the package. You will only be able to remove packages that you have added. All installed packages are shown, but the pre-installed packages cannot be deleted since they are necessary for proper operation of the node.

**Authorized SSH Keys** Uploading ssh keys allows computers to connect to a node via ssh without having to know the password. The ssh keys are generated on your computer using built-in utilities or the PuTTY program's *Key Generator*. Once you have the key files on your computer, you can upload its *public* key to your AREDN® node. If you want to remove an installed key, select it and click the *Remove* button.


---

**Note:** If you plan to use ssh keys you may want to review **Use PuTTYGen to Make SSH Keys** in the **How-To Guide** section which describes this process in detail for users of Microsoft Windows computers.

---

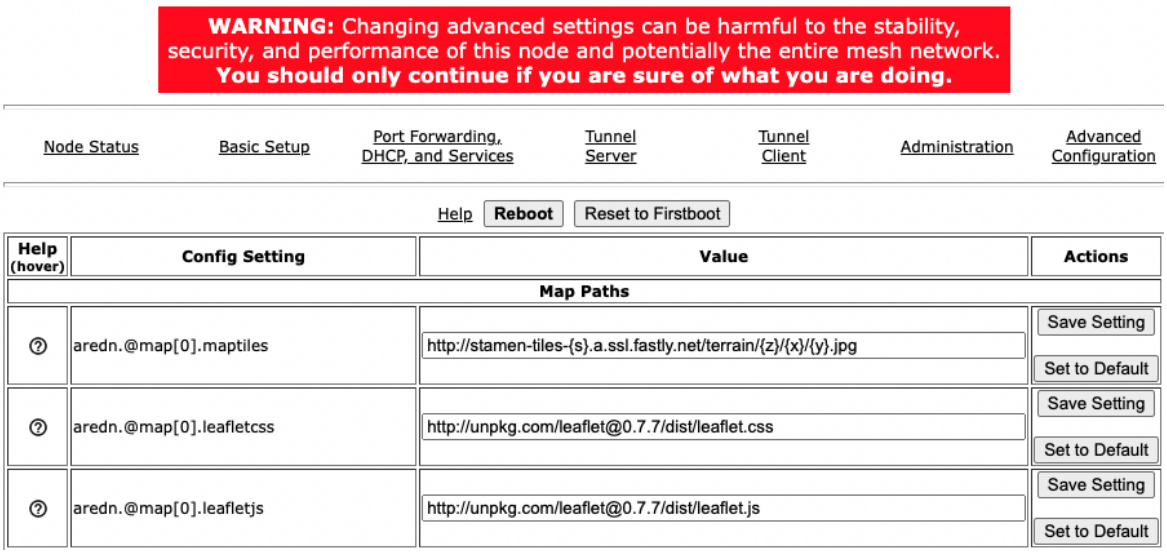
**Support Data** There may be times when you want to view more detailed information about the configuration and operation of your node, or even forward this information to the AREDN® team in order to get help with a problem. Click *Download Support Data* to save a compressed archive file to your local computer.

## 8.6 Advanced Configuration

The **Advanced Configuration** section allows you to change settings for various items that may be available on the type of hardware you are using. Not all hardware can support every value. These settings are best left as default unless you have a clear understanding of why you need to change the defaults for your node or network.

Above the settings table there are links that allow you to view the node help file, reboot the node, or reset the node to a firstboot or "NOCALL" configuration. You can edit or select a setting and then click the *Save Setting* button at the right side of the row to implement the change. You may also reset an item to default values by clicking the *Set to Default* button. Each row has hover help which can be displayed by hovering your cursor over the question mark icon at the left side of each row.

**Map Tile and Script Paths** These fields contain the external URLs for map tiles and leafletjs *css* and *javascript* files used for interactive maps.




**Firmware and Package Download Paths** These fields contain the URLs used by the node for downloading firmware and package files during upgrades. By default they point to the AREDN® downloads server available across the Internet. You can change these paths to point to a local mesh package server in order to upgrade nodes that do not have Internet access.

---

**Note:** If you plan to create a local software repository for your mesh network, review **Creating a Local Package Server** in the **How-To Guide** section.

---

| Firmware Paths | | | |
|---|---|---|---|
| ? | aredn.@downloads[0].firmwarepath | http://downloads.arednmesh.org/firmware | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_core | http://downloads.arednmesh.org/snapshots/targets/ath79/generic/packages | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_base | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/base | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_arednpackages | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/arednpackag | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_luci | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/luci | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_packages | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/packages | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_routing | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/routing | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_telephony | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/telephony | Save Setting / Set to Default |
| ? | aredn.@downloads[0].pkgs_freifunk | http://downloads.arednmesh.org/snapshots/packages/mips_24kc/freifunk | Save Setting / Set to Default |

**PoE and USB Power Passthrough** These rows will only appear in the table 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, as well as PoE power passthrough on Ethernet port 5. You are allowed to enable or disable power passthrough on nodes with ports that support this feature. Move the slider to **ON** and click *Save Setting* to enable power passthrough.

| Power Options | | | |
|---|---|---|---|
| ? | aredn.@poe[0].passthrough | OFF ON | Save Setting / Set to Default |
| ? | aredn.@usb[0].passthrough | OFF ON | Save Setting / Set to Default |

**Tunnel Server *maxclients* and Tunnel Client *maxservers*** These rows will appear in the table only if the AREDN® tunneling package is installed on your node. By default a node is allowed to host up to 10 clients in its *Tunnel Server* display and connect with up to 10 servers in its *Tunnel Client* display. The *maxclients* and *maxservers* settings provide a method for

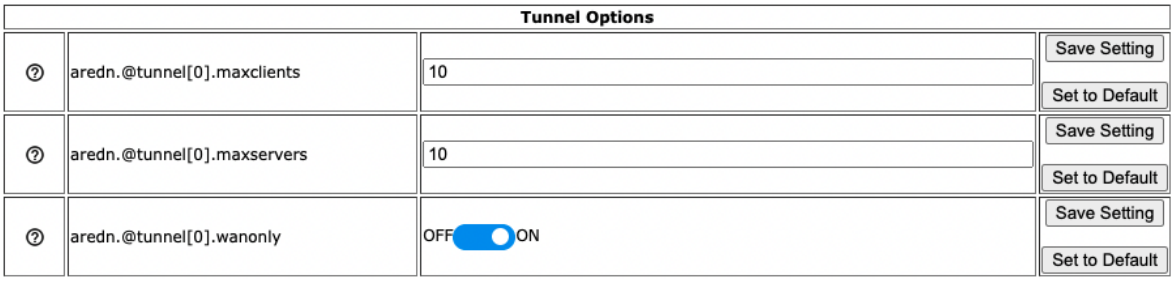
adjusting the defaults.

---

**Important:** If you plan to change these settings, review **Changing Tunnel Max Settings** in the **How-To Guide** section.

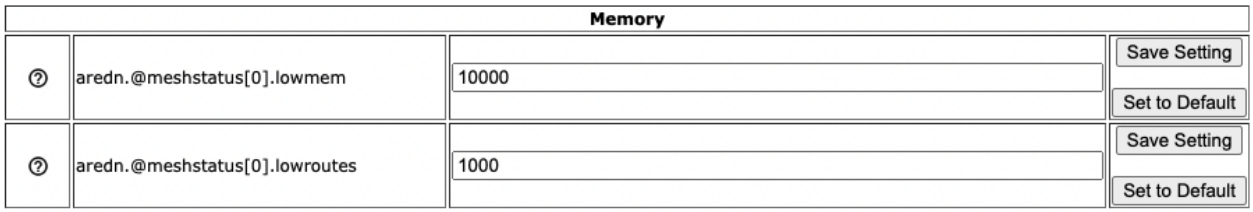
---

Use caution when increasing the *maxclients* or *maxservers* values. Enter only *zero* or positive integers up to a maximum value for the number of active connections your node hardware can handle, since each active tunnel connection consumes system resources that the node may need for normal operation.




**Tunnel WAN Only Setting** This setting is enabled by default and it prevents tunnel traffic from being routed over the Mesh RF network. It limits tunnels to using the WAN interface, which is typically the intended route. If in your situation you need tunnel traffic to be routed over RF to a node with WAN access, then you can disable this setting to allow that traffic to pass.

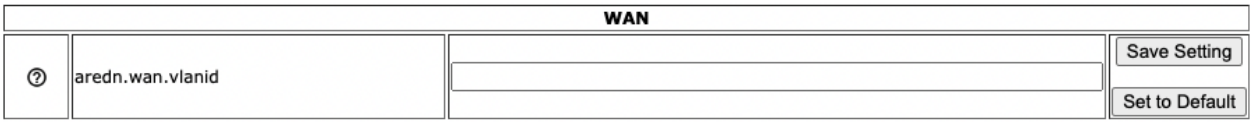
**Low Memory Thresholds** As the number of nodes increases in a mesh network, the processing requirements also increase for displaying all of the mesh routes on your node's *Mesh Status* display. For older nodes with limited memory resources, the mesh status display may become very sluggish on large mesh networks. Recent firmware improvements have made the *Mesh Status* display much more responsive, and two new **Advanced Configuration** values are available for setting the *Low Memory Threshold* and maximum number of routes to be displayed. Currently the default low memory threshold is 10,000 KB, which if reached will limit the *Mesh Status* display to the 1,000 closest routes. These values can be adjusted to lower values if your node has limited memory.



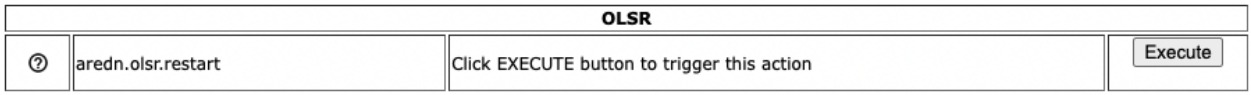
**WAN Interface VLAN Number** This feature only applies to node hardware which uses a VLAN tag for the WAN interface. It will not appear on hardware where the Ethernet ports are on a switch chip, since changing the default VLAN number is not supported on those devices at the present time. It will appear as a `blank` field on devices that have a dedicated WAN port and therefore do not need a VLAN tag for their WAN interface.

If you have node hardware that uses a VLAN tag for the WAN interface, then the default WAN VLAN identifier is 1. In some cases this default VLAN may be in use already or may be reserved by other equipment on your network. This field allows you to change the VLAN number being used on your node's WAN interface.

> **Caution:** If you plan to change this setting, do not use the number 2 (which is reserved for the DtD VLAN on AREDN® nodes) or 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.




**OLSR Restart** The OLSR (Optimized Link State Routing) process can be restarted when you want your node to rebuild its mesh routing table but you do not want to do a full reboot. Click the *Execute* button to restart OLSR.




There is a known intermittent issue that may occur when a node boots. If OLSR fails to propagate information or does not receive all the network hostnames, a one-time restart of OLSR should resolve the issue. OLSR should be restarted on your node if other nodes' *Mesh Status* display have your node's IP address rather than hostname or if "dtdlink" or "mid" is shown in your node's hostname on their *Mesh Status* display. If your node's *Mesh Status* display shows the IP address rather than hostname for a remote node, then that remote node should restart OLSR.

**AREDN Alert Message (AAM) Refresh** The AREDN® development team may post messages

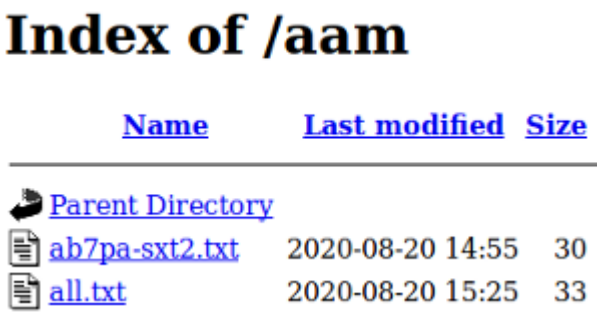
which Internet-connected nodes can automatically download. You can execute the *aam.refresh* action if you want your node to retrieve any new messages without having to wait for the next auto-refresh window. Click the *Execute* button to trigger an immediate message retrieval. This will retrieve all alerts eligible for display on your node, whether they come from the AREDN® server over the Internet or from a local message source on your mesh network.

| AREDN Alerts | | | |
|---|---|---|---|
| ⊘ | aredn.aam.refresh | Click EXECUTE button to trigger this action | Execute |
| ⊘ | aredn.@alerts[0].localpath | http://ab7pa-box2/aam | Save Setting / Set to Default |
| ⊘ | aredn.@alerts[0].pollrate | 1 | Save Setting / Set to Default |
| ⊘ | aredn.aam.purge | Click EXECUTE button to trigger this action | Execute |

**AREDN Alerts Local Path** This field allows you to enter the URL for a local alert message repository. If you configure such a local repository then your nodes without Internet access can also receive alert messages pertinent to your local mesh. Enter the URL without a trailing backslash.

A local message repository should be configured on a mesh-connected web server which allows nodes to query the URL you entered. No Internet access is required for this feature to work as designed. You can consult with your local web server administrator in order to obtain the correct URL for the local message repository. Use the following file naming convention on the web server:

- Create text files for individual nodes by using only lowercase characters with the exact node name, followed by the `.txt` extension as shown below.
- 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.

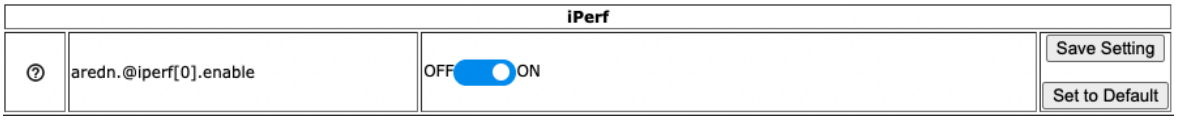


It is possible to include HTML tags in your message text, such as using the `<br />` tag to display subsequent text on the next line. However, it is best practice to keep alert messages short in order to minimize the height of the alert banner displayed on node webpages.

**AREDN Alert Pollrate** This field allows you to set the polling rate or interval in hours at which the node will check for message updates. The default polling rate is once every 12 hours, but you can make this value smaller if you want your node to check for updates more frequently.

**AREDN Alert Message Purge** Use this purge setting if you want to immediately remove the AREDN® Alert Message banner from your node. Click the *Execute* button to trigger an immediate message banner removal. This will remove all alert messages, whether they originated from the AREDN® server over the Internet or from a local message source on your mesh network.

**iperf URL Feature** The *iperf URL* feature is described in the "Test Network Links with iperf3" section of the **How-To Guide**. It is enabled by default, but if you do not want your node to participate in such remote iperf tests then you can disable its ability to respond to those queries using this setting. Move the slider to `OFF` and click *Save Setting*.




## 8.7 Node Reset Button

The reset button on an AREDN® node has two built-in functions based on the length of time the button is pressed.

With the node powered on and fully booted:

- **Hold for 5 seconds to reset the password and DHCP service**
- **Hold for 15 seconds to return the node to "just-flashed" condition**

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

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

NINE

# NETWORKING OVERVIEW

This **Network Design Guide** will discuss some of the useful principles for creating robust data 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 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 OLSR (Optimized Link State Routing protocol) 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.

Link: AREDN Webpage

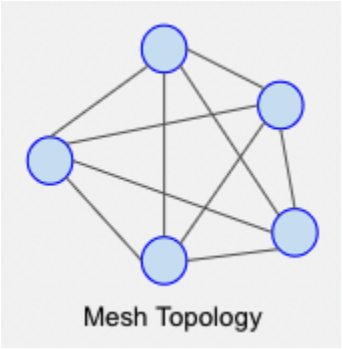
Link: AREDN Webpage

CHAPTER

TEN

# 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.



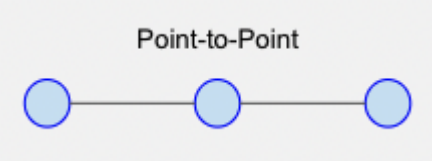
Mesh Topology

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.

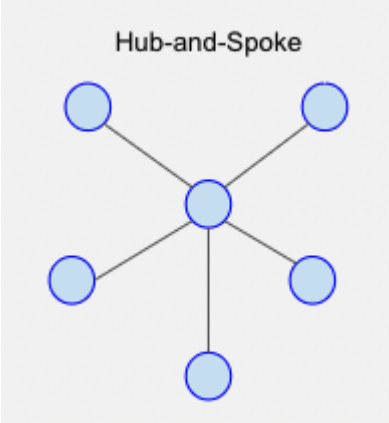
## 10.1 Types of Topologies

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.

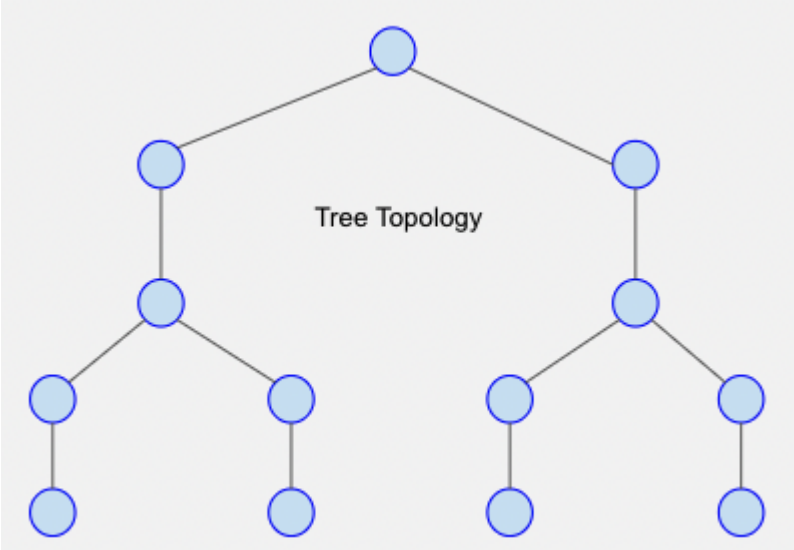
**Point-to-Point Topology** Point-to-Point topologies are best suited for moving data between the far endpoints, potentially using one or more intermediate nodes in order to traverse different types of terrain or to overcome obstacles in the network path.




**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.

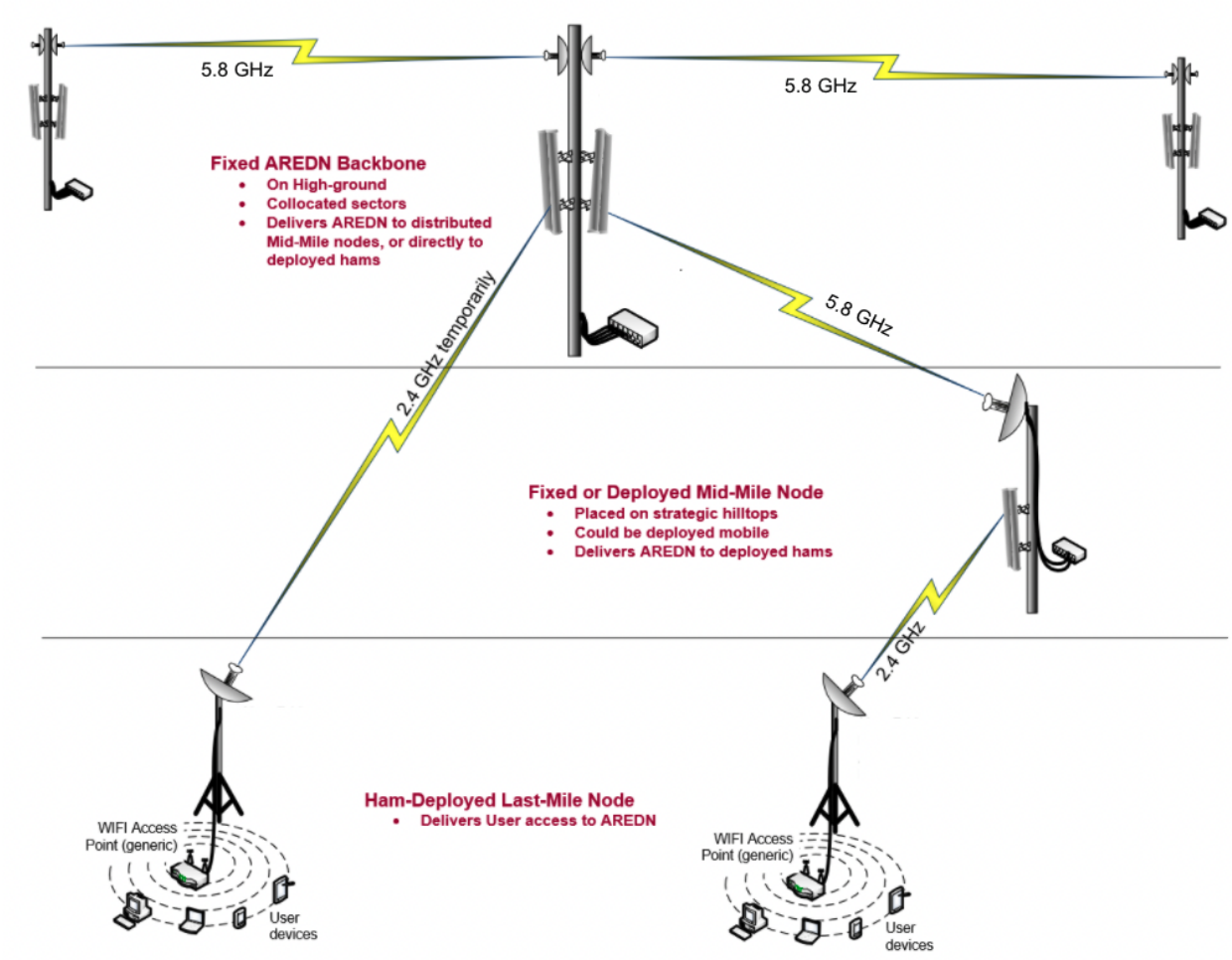



**Tree Topology** A tree topology can be used to segment or partition network traffic, keeping specific data within a localized area while also allowing for links to remote parts of the network. The tree topology uses a parent-child hierarchy to structure the paths that data can take. This design can 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.




## 10.2 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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

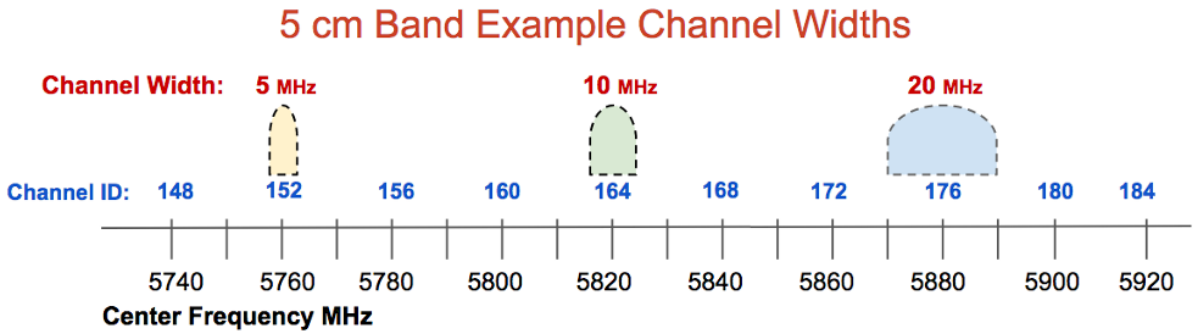
# ELEVEN

# 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.

**Caution: 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 channels that are available for AREDN® networking in the US.

| Band | Frequency Range | Channels |
|---|---|---|
| 33 cm | 902-928 MHz | 4 |
| 13 cm | 2390-2450 MHz | 10 |
| 9 cm | 3300-3445 MHz | 14 |
| 5 cm | 5650-5925 MHz | 54 |

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.

## 11.1 900 MHz Characteristics

**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 in all but the most remote areas the RF noise floor may be very high, which reduces the SNR and results in packet loss, retransmission delays, and lower usable link quality.

Another disadvantage is that the equipment can be more expensive than devices that operate in the 2.4 and 5.8 GHz bands. Also the entire band is quite narrow (25 MHz) which means that only one, two, or four radio channels can exist in this shared frequency range, depending on the channel width that is selected.

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

**Advantages** The advantage of this frequency band is that its longer wavelength makes it better suited for penetrating some types of obstructions and foliage which would normally block signals at higher frequencies. Its NLOS (Non Line of Sight) propagation characteristics may be exactly what is needed in order to establish an RF link between two difficult locations.

## 11.2 2.4 GHz Characteristics

**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 10 MHz each.

One concern with all of the higher 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. For example, on a link in the 13 cm band with 10 miles between nodes, the first Fresnel Zone radius will be 72 feet. 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. 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.

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

* Only 5 MHz channel width is available on channel 8

**Advantages** 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.

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-tuned antennas, 2.4 GHz signals can propagate across very long distances.

## 11.3 3.4 GHz Characteristics

**Disadvantages** As mentioned above, 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 above. However, the 60% no blockage radius is still about 37 feet. Consider node AGL and terrain in order to minimize obstructions.

Equipment for the 9 cm band is less readily available and is typically more expensive due to less consumer demand. 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. Also, 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. Consider this before investing in or implementing new AREDN® devices in this band.

| 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 | Shared with US non-Amateur users | | | | | | | | | | | | | |

| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
|---|---|---|---|---|---|---|---|---|---|
| 3.450 | 3.455 | 3.460 | 3.465 | 3.470 | 3.475 | 3.480 | 3.485 | 3.490 | 3.495 |
| ~~ Elimination in US by 14 April 2022 ~~ | | | | | | | | | |

**Advantages** The main advantage for using the 9 cm band is that it has more available bandwidth for use in unshared channels than any other band. 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 will increase the SNR to improve signal quality if that is an issue for a particular link. Equipment in the 9 cm band is well-suited for *Backbone Links* since there is less possibility for interference from other devices sharing these frequencies at tower sites. With clear line of sight and well-tuned antennas, 3.4 GHz signals can propagate across very long distances.

## 11.4 5.8 GHz Characteristics

**Disadvantages** As mentioned previously, there must be clear line of sight and the Fresnel Zone between nodes also must be unobstructed. 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 above. 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.

| 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 (max EIRP 1000mW) | | | | | | | | | | | | | | | 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 (max EIRP 200W) | | | | | | | | | | | | | | | | | |

| 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 (max EIRP 200W) | | | | | | | | | *Shared with Intelligent Transportation System* | | | | |

Power limits shown are for non-Amateur services which share the specified channels.

**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 will increase the SNR to improve signal quality if that is an issue for a problem 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-tuned antennas, 5.8 GHz signals can propagate across very long distances.

---

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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

TWELVE

# 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.

## 12.1 Wireless Network Operation

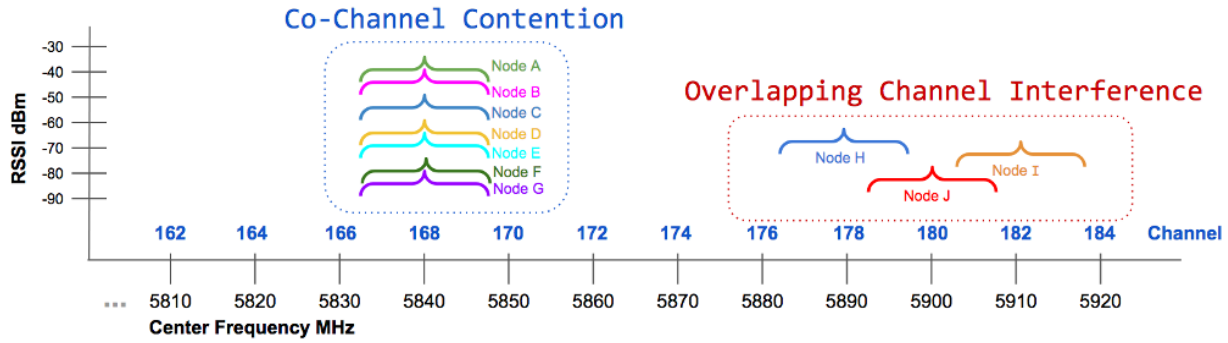
A wireless network is a shared half-duplex medium on which only one station at a time should transmit. In that sense wireless operations are analogous to other types of radio transmissions. If two stations key up their transmitters at the same time, they will interfere with each other to the extent that 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 OLSR (Optimized Link State Routing protocol) to maintain routes for the network as a whole, but this OLSR 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.

### 12.1.1 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.

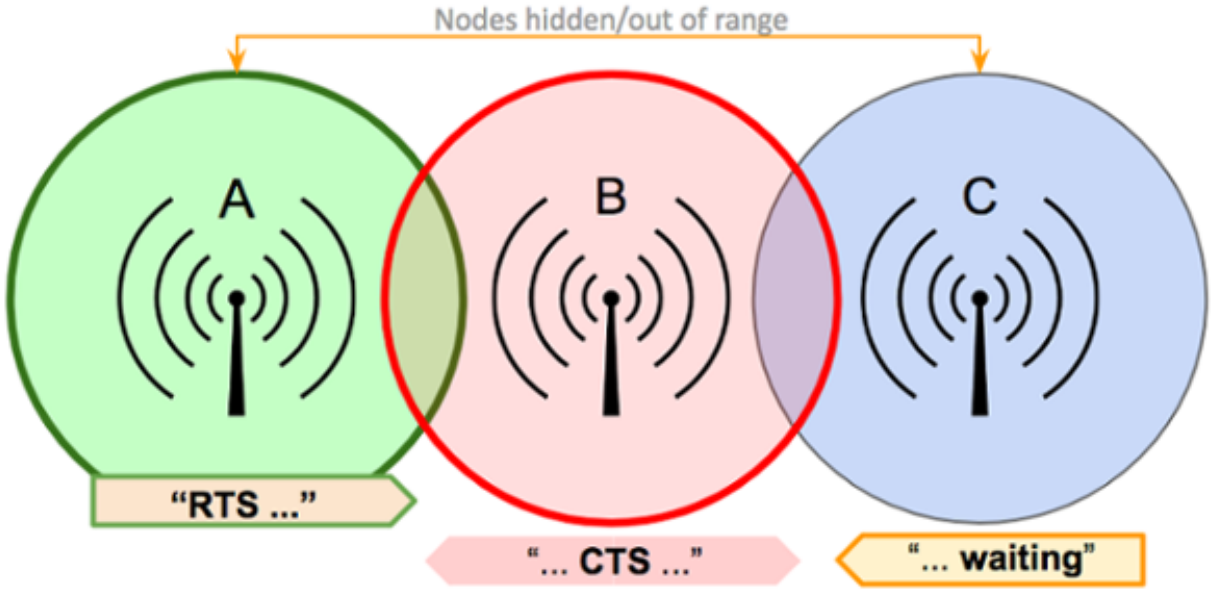
This example is not meant to show that having only seven nodes will overload a channel. There is no established rule of thumb for channel sharing that specifies how many nodes are too many. The answer depends on the number of nodes, the bandwidth in use to support required services, the link signal qualities, 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.

### 12.1.2 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 example diagram, **A** can hear **B** but cannot hear **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 appear to have a clear channel for transmission.




Request to Send / Clear to Send (RTS/CTS) messages are used by AREDN® nodes to minimize or eliminate this issue. For example, node **A** broadcasts a short RTS message with a proposed timeslot/duration for transmitting its complete 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.

Two other approaches may also alleviate the hidden node issue. You may be able to make the hidden nodes visible to each other, for example by increasing their signal strength. The alternative is to isolate the nodes completely by placing them onto different bands or channels. 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. It may be more appropriate to create a separate link between the sites and to put the radios on a different band or channel.

Another case is when there is one poor quality link over which all traffic must be routed. The handshaking and data retransmissions may cause all the other nodes to wait. The entire network can be impacted by one low quality path which becomes a single bottleneck. If at all possible you should increase the signal quality of that vital link or install a better link as an alternate path.

### 12.1.3 Route Flapping

This is another issue that can lead to performance problems on a network. You may have parallel paths between two *Remote Nodes*, and these paths have similar ETX values which indicates that the cost of using either route is comparable. These two paths may appear to be functioning well most of the time.

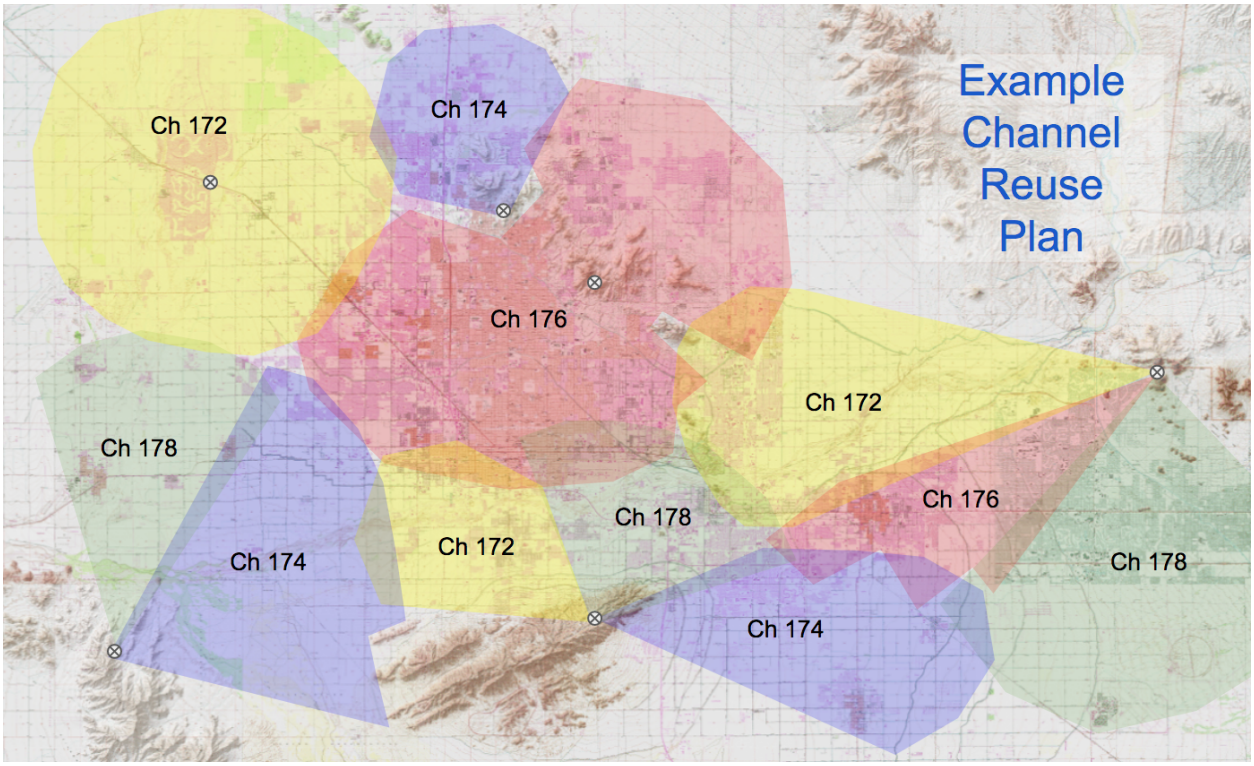
However, when a bandwidth-intensive application such as video conferencing begins sending traffic across one of the paths, you may notice that link getting bogged down and the ETX will drop below that of the other path. At this point OLSR switches to the alternate path which now has a lower cost. The video stream then bogs down its new path, which lowers the ETX, and OLSR switches back to the original link whose ETX is better again. This situation may continue indefinitely, with neither path being able to deliver the traffic adequately.

This issue can happen on multi-hop links with similar ETX which seem to work fine until they are loaded with traffic. Then packet loss begins to occur, connections time out, and neither path is reliable during that cycle. One solution might be to improve the multi-hop link cost by increasing the signal quality of the links along one of the paths. Conversely, you could also turn down the power on the alternate path to increase its cost. If bandwidth-intensive traffic must be passed between two remote endpoints, the best approach would be to design a more robust path between those two endpoints to meet that need.

## 12.2 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.

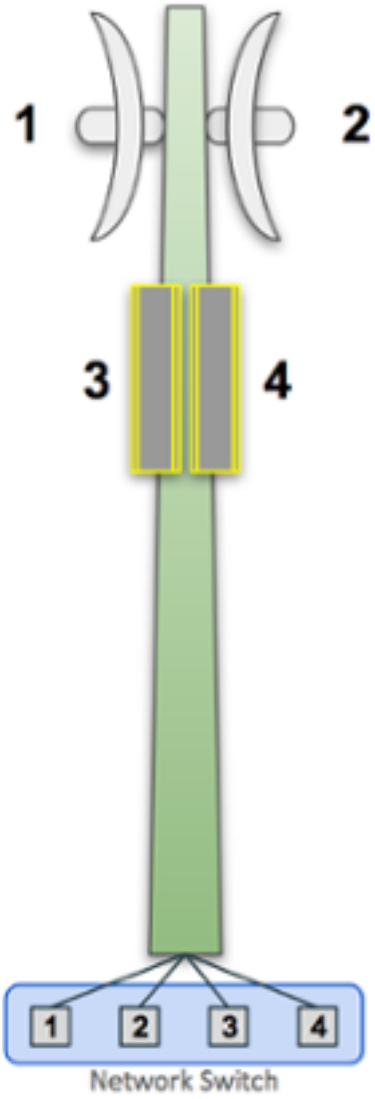
## 12.3 Collocated Nodes

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.

### 12.3.1 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. OLSR will assign a very low "link cost" (0.1) to the DtD connection and automatically route traffic between the nodes over Ethernet rather than causing the RF channel to become busy.

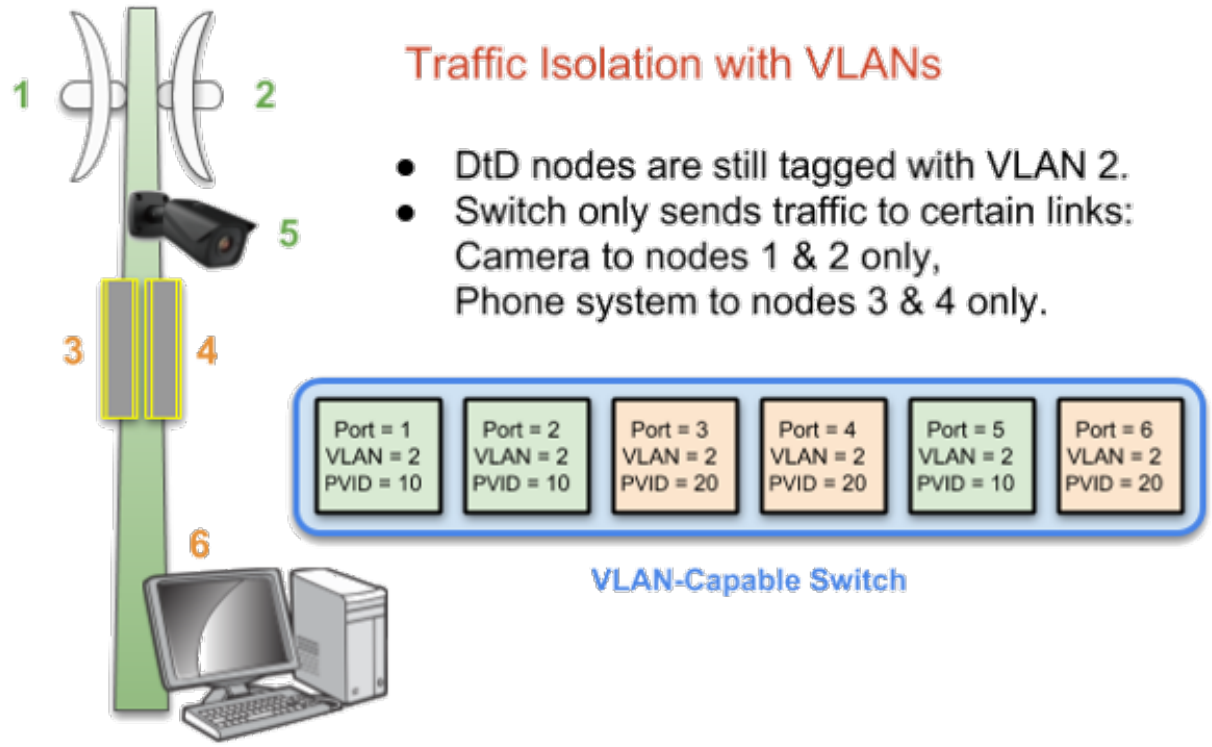



One added benefit of DtD linking is that you can link nodes which are operating on different bands and channels. Nodes that are using *Channel Separation* to segment traffic can still pass data at high speeds through their DtD link and be members of a single network. At a tower site like the one shown 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. OLSR will 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 it on its DtD link.

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 (Voice over IP) 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.

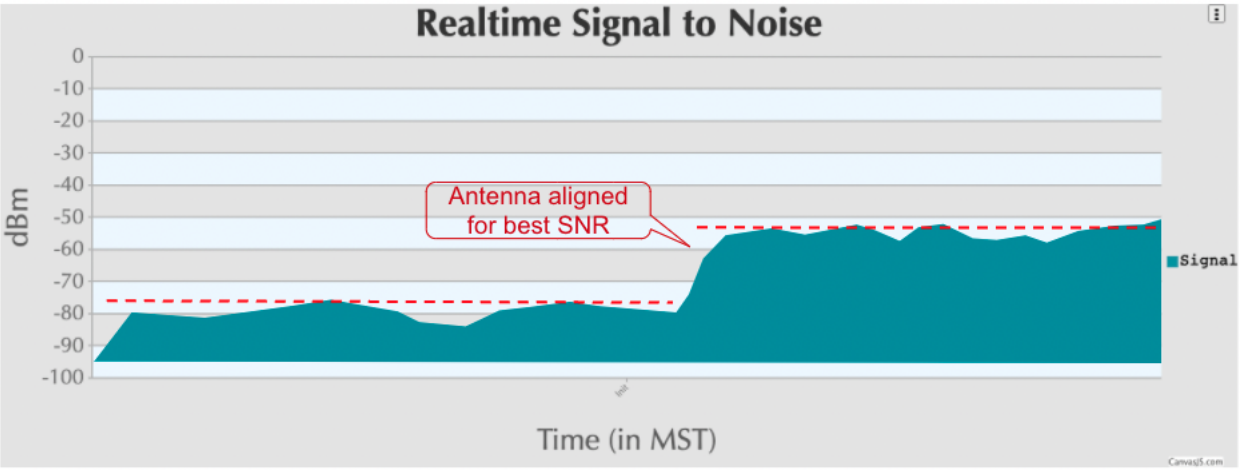



### 12.3.2 Antenna Polarization

Most of the latest AREDN® devices use dual polarity antennas and MIMO 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.

### 12.3.3 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.




## 12.4 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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# THIRTEEN

# 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.

## 13.1 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:

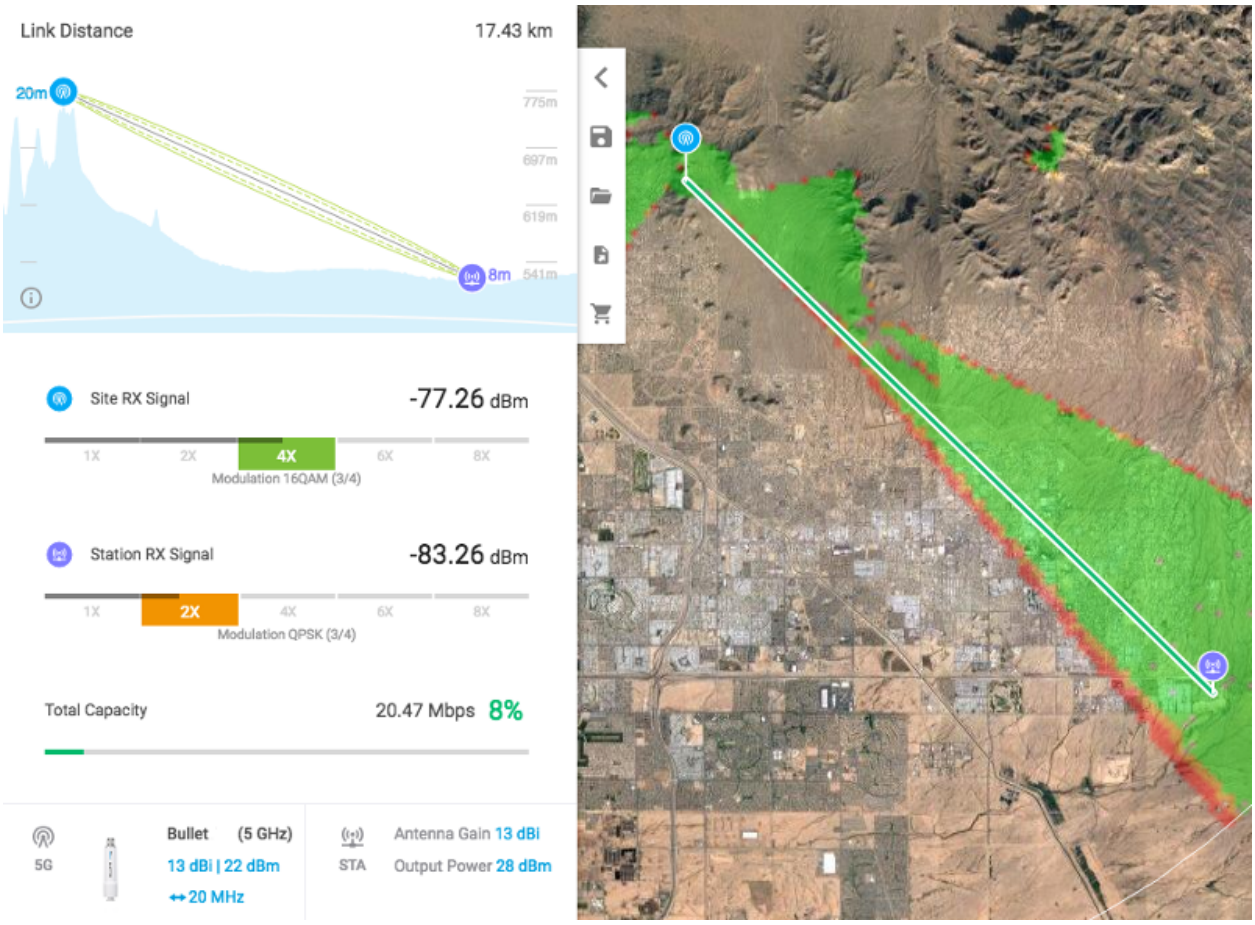
- Latitude and Longitude
- Antenna AGL
- Frequency
- Transmit Power
- Line Loss
- Antenna Gain
- Receiver Sensitivity

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

### 13.1.1 Ubiquiti AirLink Tool

If you are using Ubiquiti radios there is a free modeling tool available on the Ubiquiti website (http://link.ubnt.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.

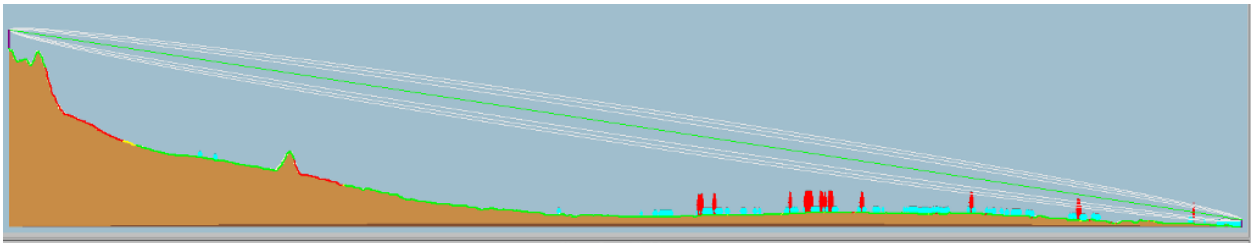
The path profile is color coded to indicate whether the link quality is adequate. It displays the link distance, line of sight, as well as the Fresnel Zone and 60% clearance area. It also estimates the signal levels at each endpoint and the predicted throughput for the link. An example *AirLink* path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



### 13.1.2 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 ° |

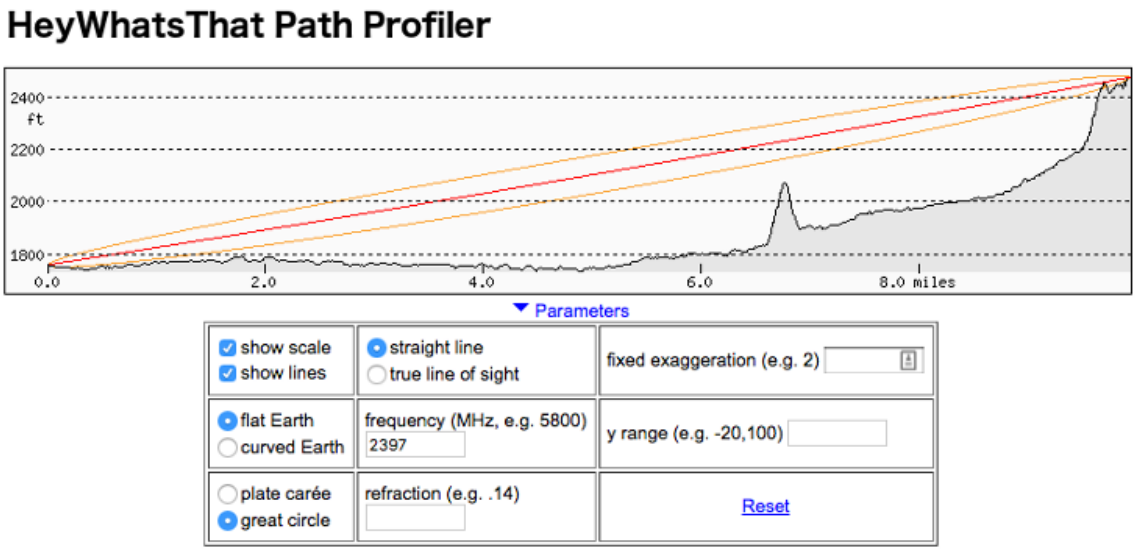
| **Radio system** | | **Propagation** | |
|---|---|---|---|
| TX power | 28.45 dBm | Free space loss | 132.52 dB |
| TX line loss | 0.50 dB | Obstuction loss | -3.16 dB |
| TX antenna gain | 20.00 dBi | Forest loss | 0.00 dB |
| RX antenna gain | 16.00 dBi | Urban loss | 1.00 dB |
| RX line loss | 0.50 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** |

### 13.1.3 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.




### 13.1.4 Radio Fresnel Tool

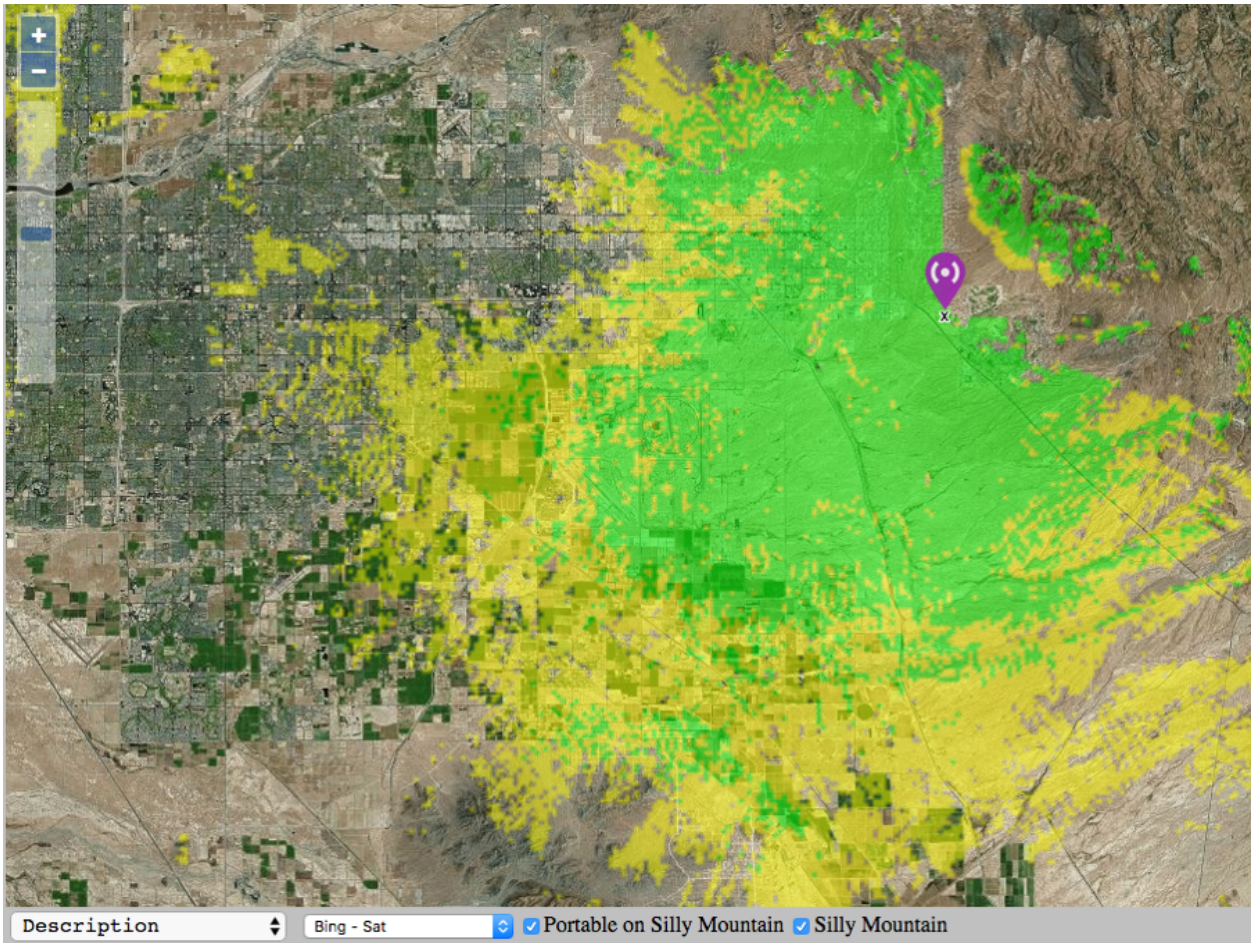
This web-based tool will generate a KML file which can be viewed as a 3D path profile using *Google Earth* software. Radio Fresnel is available here: http://www.radiofresnel.com

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.


## 13.2 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.




Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# FOURTEEN

# 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. 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.

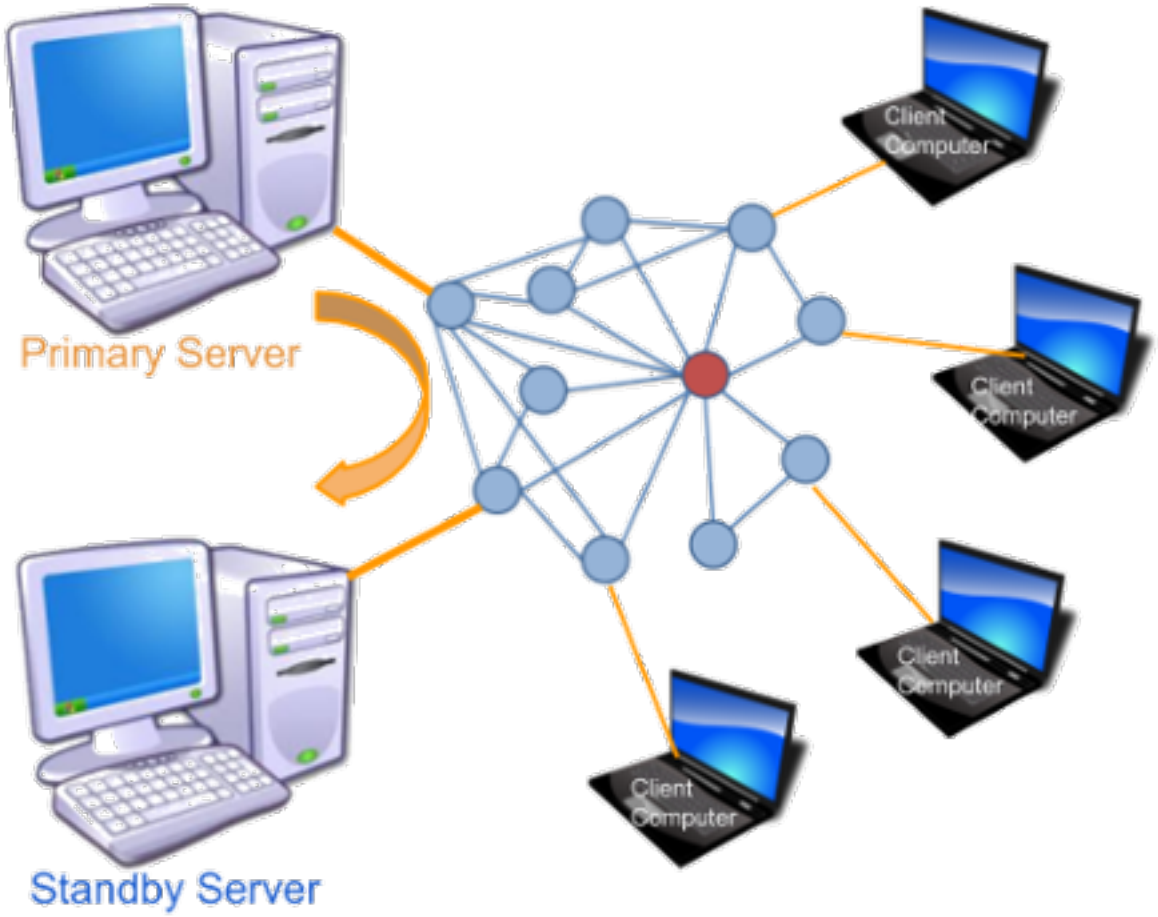
**Keeping Multiple Servers in Sync**

Since the application *server* must be reachable on the network in order for *clients* to function, and since a solitary server can be a single point of failure, it may be useful to explore ways for redundant servers to be kept in sync across the network. If one server becomes unreachable, a backup or failover server could be used to keep the service running.

For mission-critical services on high speed data networks, *Disaster Recovery* designs are often implemented to ensure that services continue operating in the event of a failure. There are several methods for accomplishing this, which usually involve duplicating server hardware and software with some type of data replication between these systems. At a high level, two basic designs could be implemented as described below.

***Manual Failover Design*** In this design there is a primary server that remains active, with a duplicate backup server located on another network segment. The standby server is brought online only if the primary server becomes unreachable. Application data on the primary server could be copied periodically to the standby server using an intelligent utility such as rsync running as a scheduled task which copies only what has changed since the last check. This design provides a fallback that can be used in case of emergency, but it requires some degree of manual intervention to bring up the standby service on the network when the primary becomes unreachable.

***Automated Failover Design*** High Availability technology allows two or more sets of computing resources to send heartbeat signals for detecting whether their services are available across the network. Several types of open source and commercial clustering packages are available, which provide varying degrees of complexity and recovery capabilities. Suffice it to say that many options are available for ensuring the availability of mission-critical services on your network. Feel free to research, investigate, and test several of these options if you have a pressing need for highly available mesh services.




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.

---

**Summary**

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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

FIFTEEN

# 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.

## 15.1 MeshChat

MeshChat has become the primary chat service for AREDN® networks because it was written by Trevor Paskett K7FPV specifically for mesh communication. Users access MeshChat via web browser, and the service can run on the mesh node itself or preferably 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 separating 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 appended to 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 at any time. If

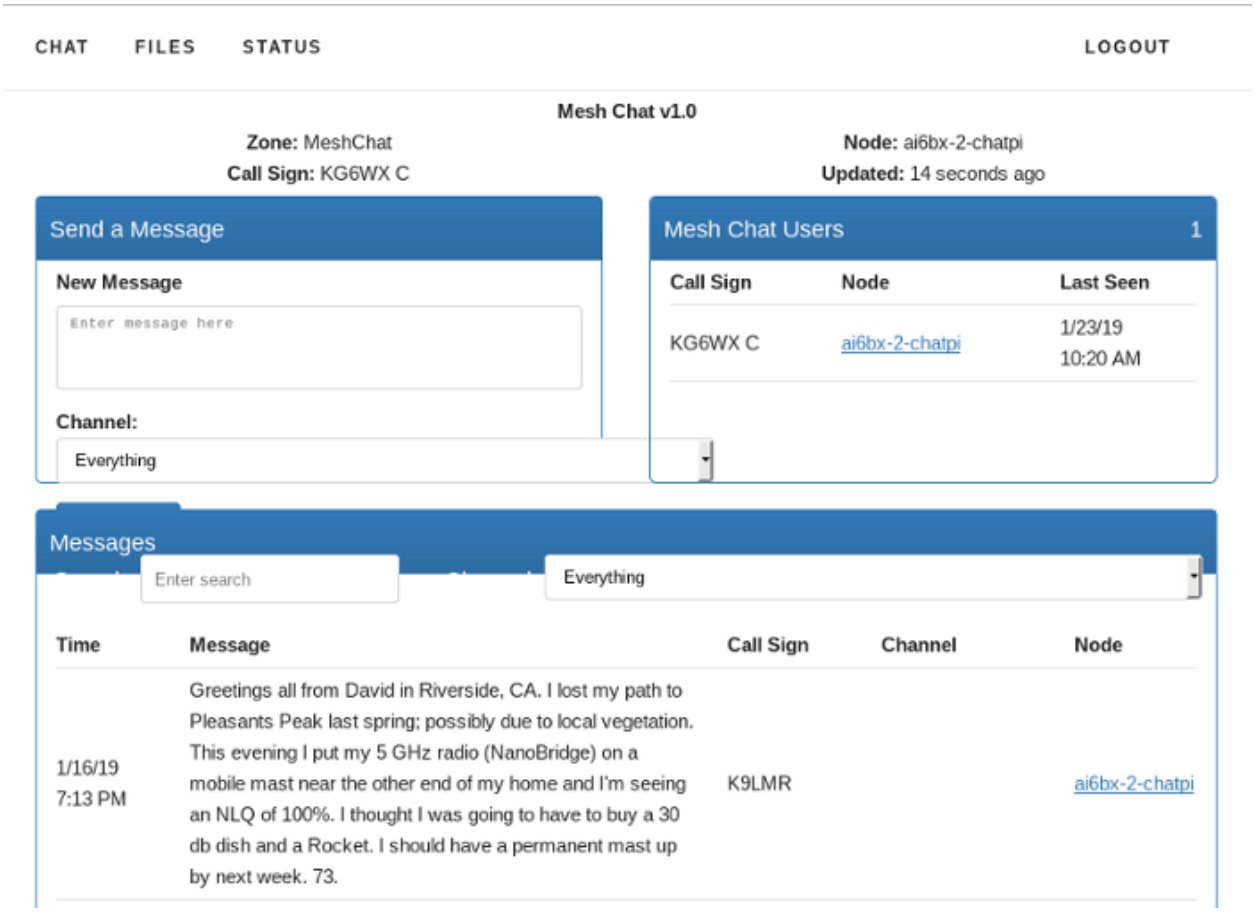
MeshChat is running on a radio node then the file storage is very limited, but if running on an external computer the file storage is limited only by the size of the disk that is allocated for MeshChat files.

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 uses the Perl programming language and is able to run either on an AREDN® node or on a LAN-connected Debian or Raspberry Pi computer. With the project to retire Perl on AREDN® nodes, there are now alternative MeshChat packages which use the Lua programming language for running on nodes. If you are running the original Perl version on an external computer, you can still use the new Lua API on your node to provide the computer with the list of MeshChat nodes. These Lua packages are available at the following links:

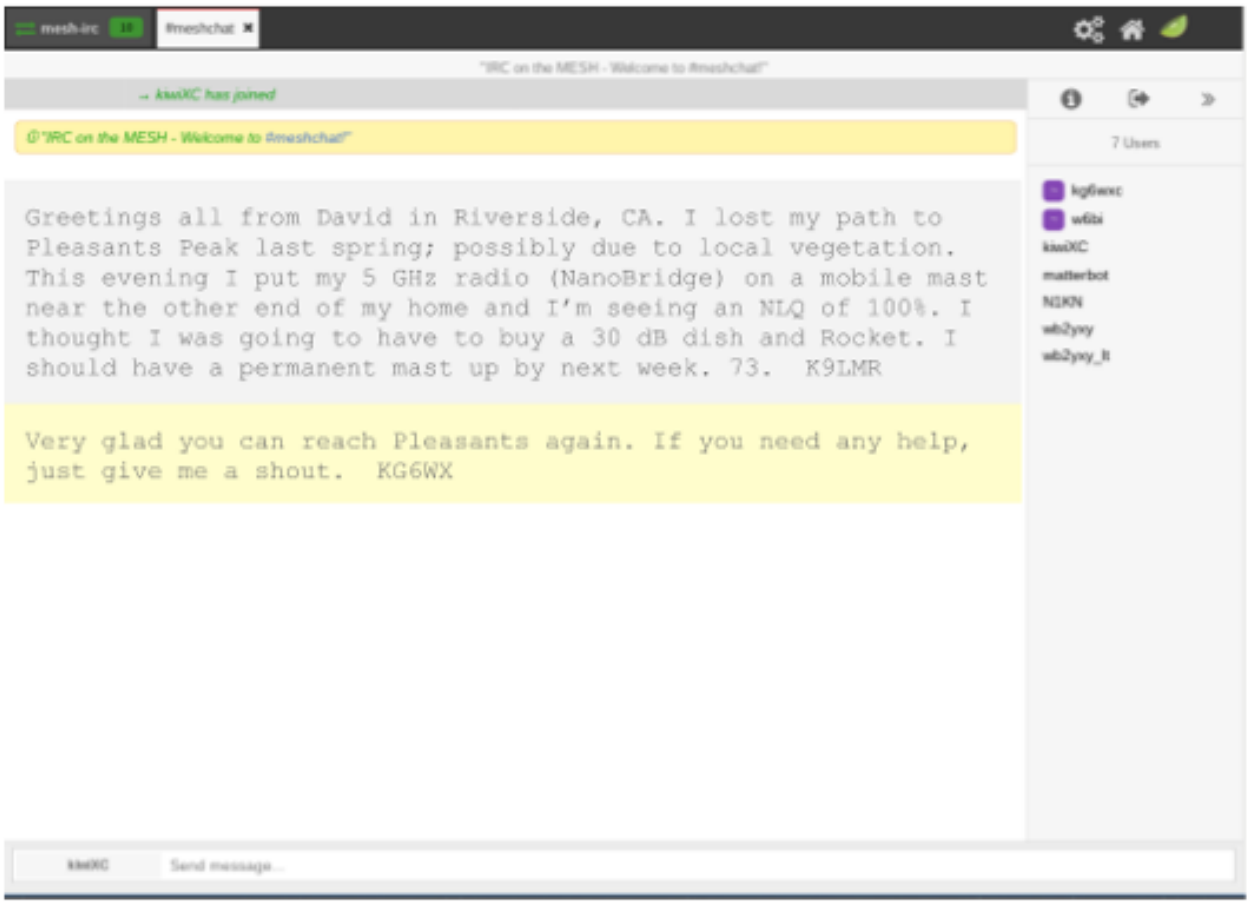
- Full MeshChat package for a node
- MeshChat package containing the node API only



## 15.2 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.

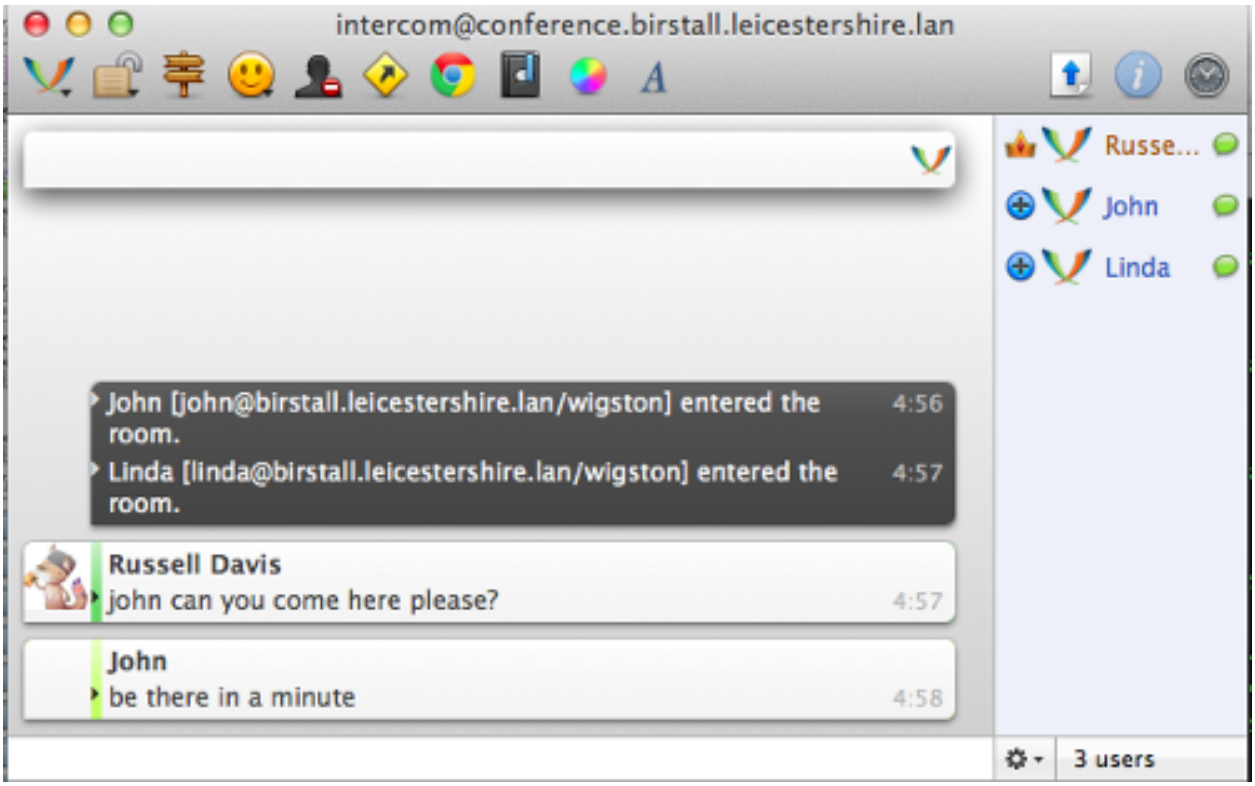
A wide variety of features and functions are available with these and similar chat programs, including various zones, channel types, and user roles. For additional information about IRC services, visit IRC Clients



## 15.3 Jabber/XMPP

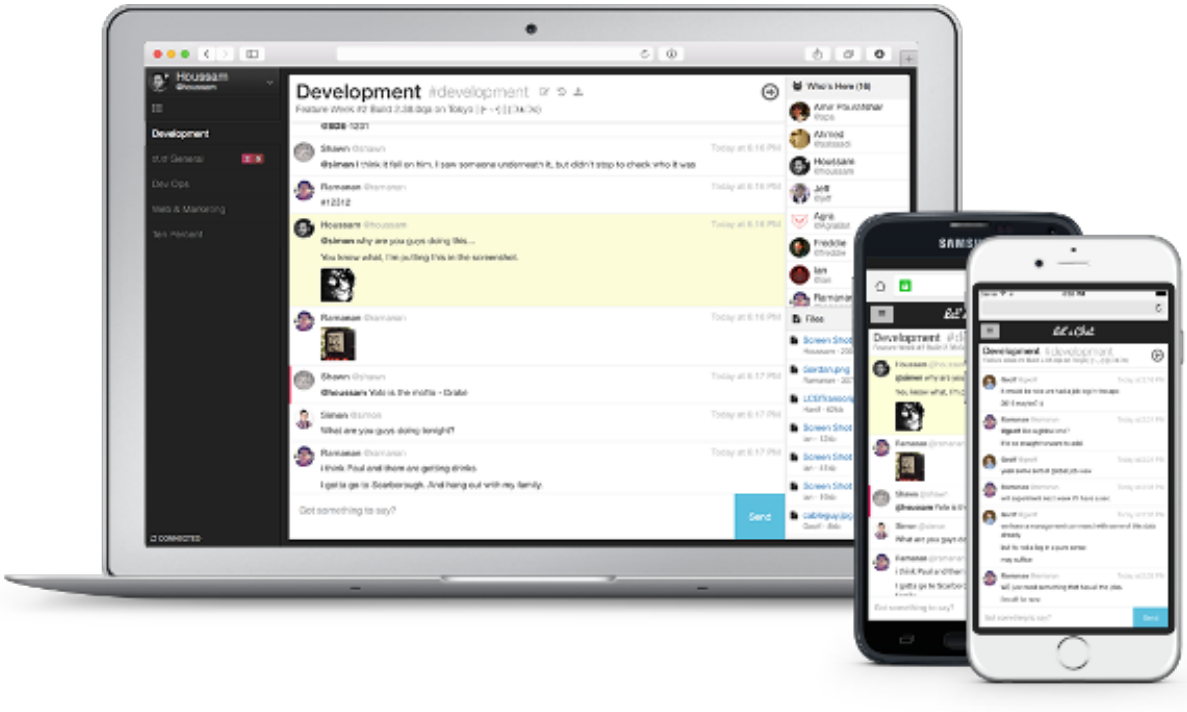
Originally known as Jabber, XMPP servers have been around for a long time but are fully compliant with modern messaging standards thanks to a large community of developers worldwide. These servers provide one-to-one messaging as well as group chat sessions. User lists have activity and presence indicators, 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




## 15.4 Let's Chat

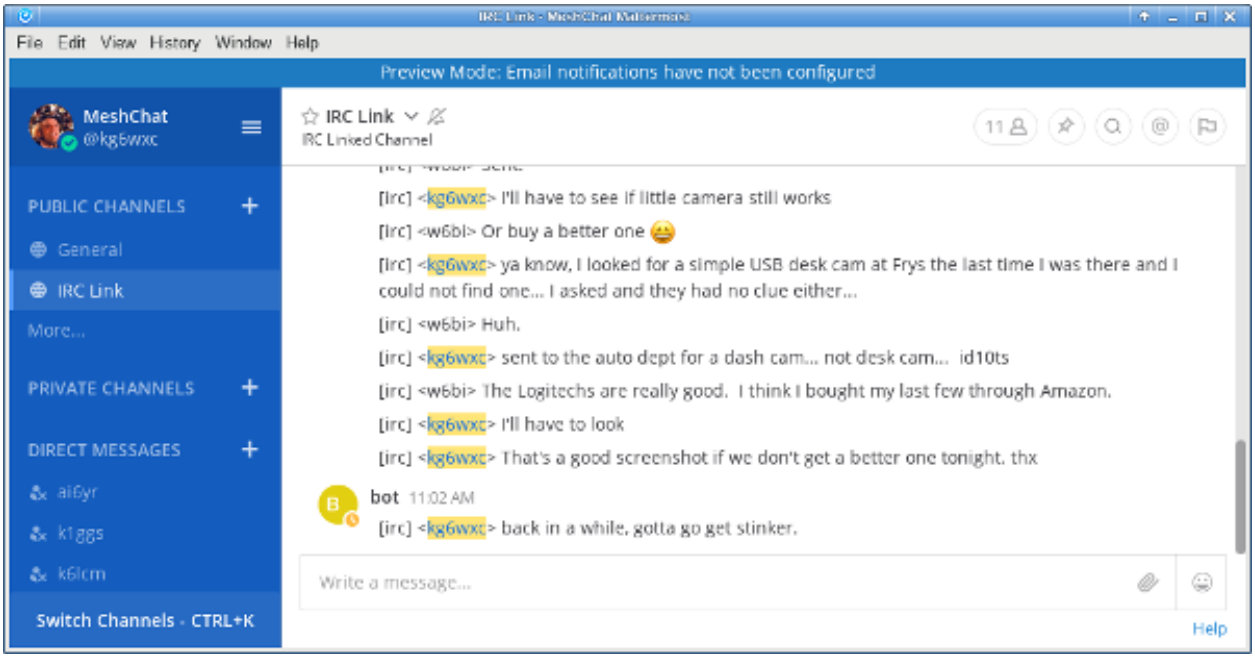
*Let's Chat* is an open source messaging service for small teams. It provides one-to-one communication between XMPP users as well as group messaging and @mentions in a variety of chat rooms. Searchable conversation history is available, in addition to text and image pasting, user activity notifications,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




## 15.5 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.

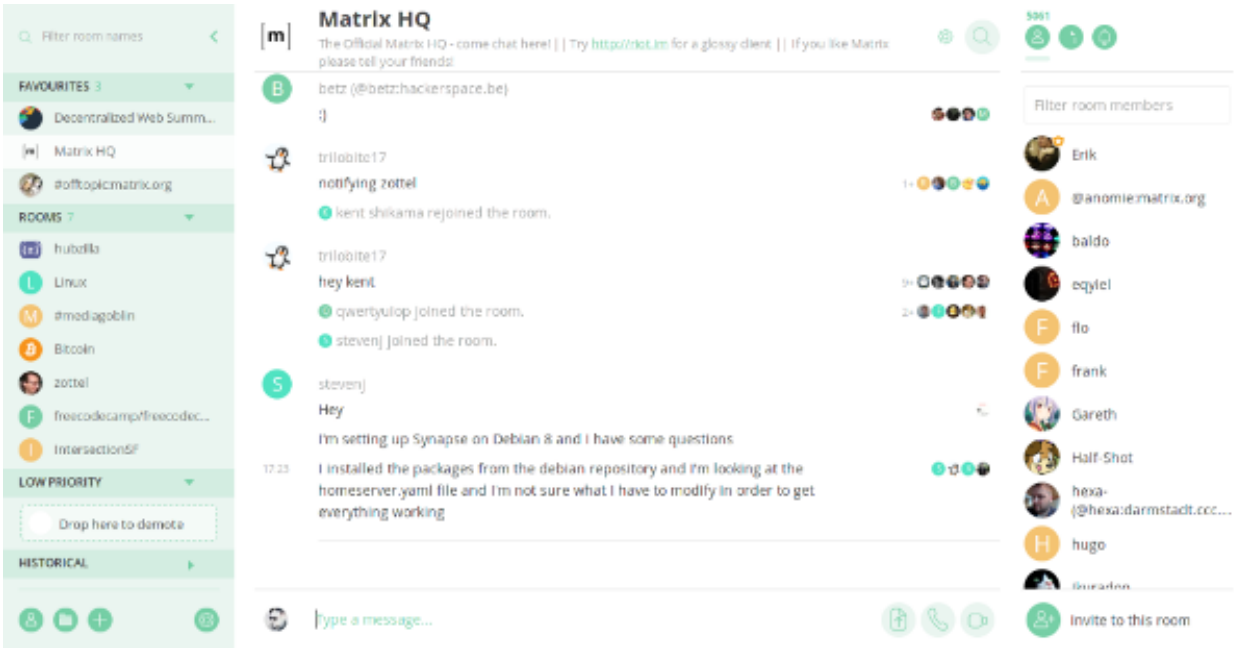
Mattermost supports @mentions, and channels are available for organizing conversations which can be topic-based, group-based, or event-based. Notifications indicate user presence and activity. File sharing is provided for PDF and text files, as well as audio, video, and image files. For additional information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




## 15.6 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




## 15.7 Example Chat Service Comparison

**Platform abbreviations:** win=MS Windows, mac=Apple, lin=Linux, rpi=Raspberry Pi

| Program | Architecture | Network Load | Age | Platform | Effort |
|---|---|---|---|---|---|
| MeshChat | mesh aware | small | new | node/rpi | 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 |

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

SIXTEEN

# 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.

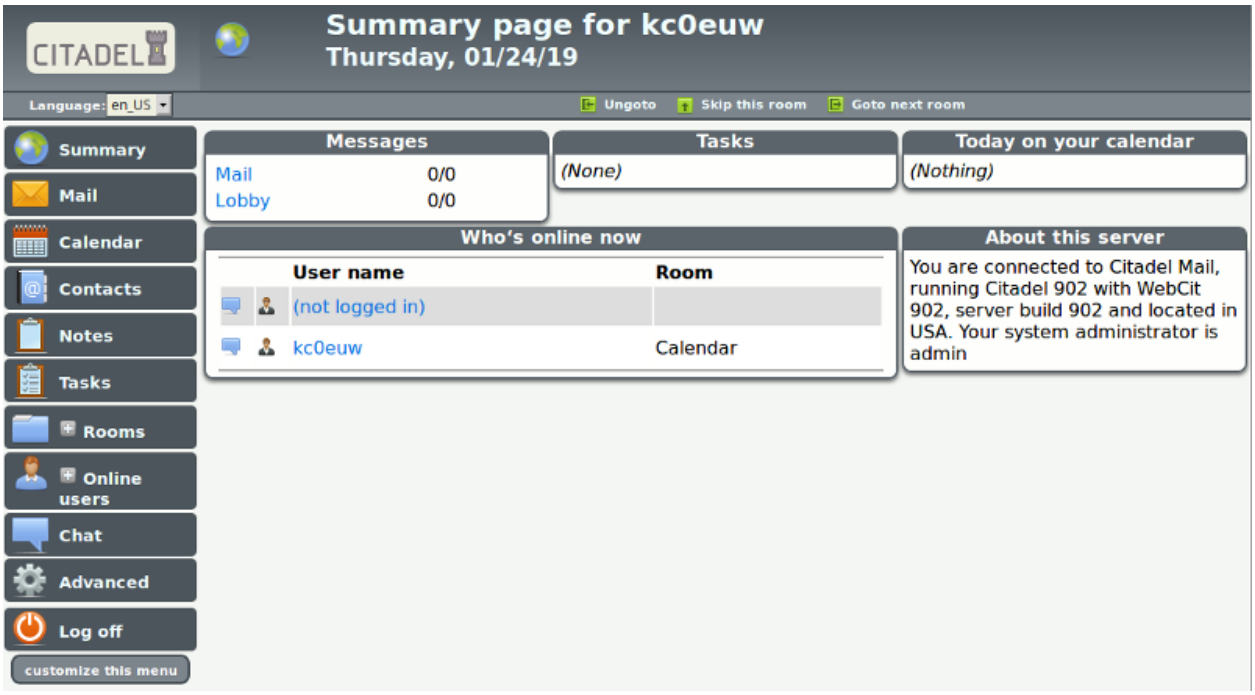
As with any client-server program, the email server must be reachable on a network segment with adequate bandwidth in order for the clients to exchange messages. If you have a choice, put your email server on one of your largest and most reliable network segments. Refer to this link for a comparison 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.

## 16.1 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




## 16.2 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.

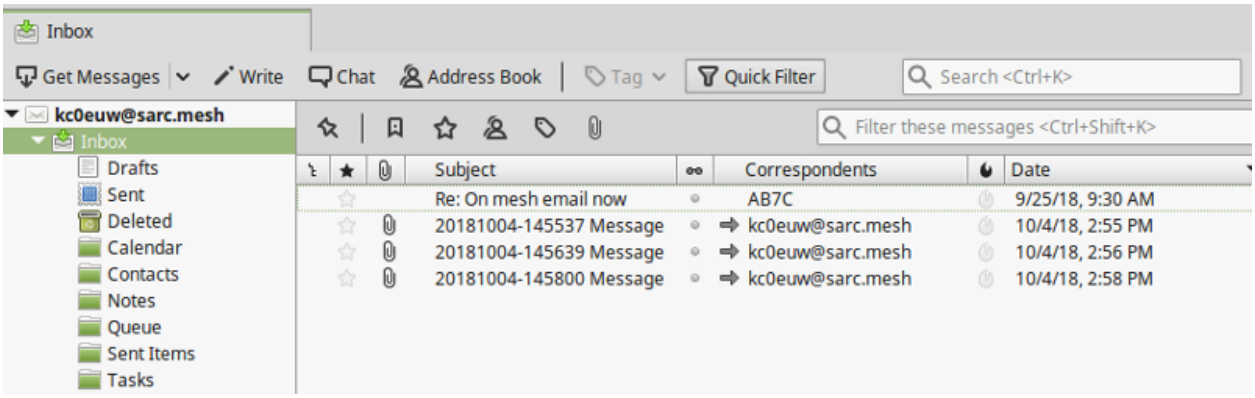
- Sendmail is the original legacy SMTP server that is still used today, although one of the newer programs below is often chosen for its ease of configuration and added security features.
- Exim is the default SMTP server in Debian Linux, is well-documented, having many configurable features, and it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

- Postfix is the default SMTP server in Ubuntu Linux and MacOS, with many integration 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.

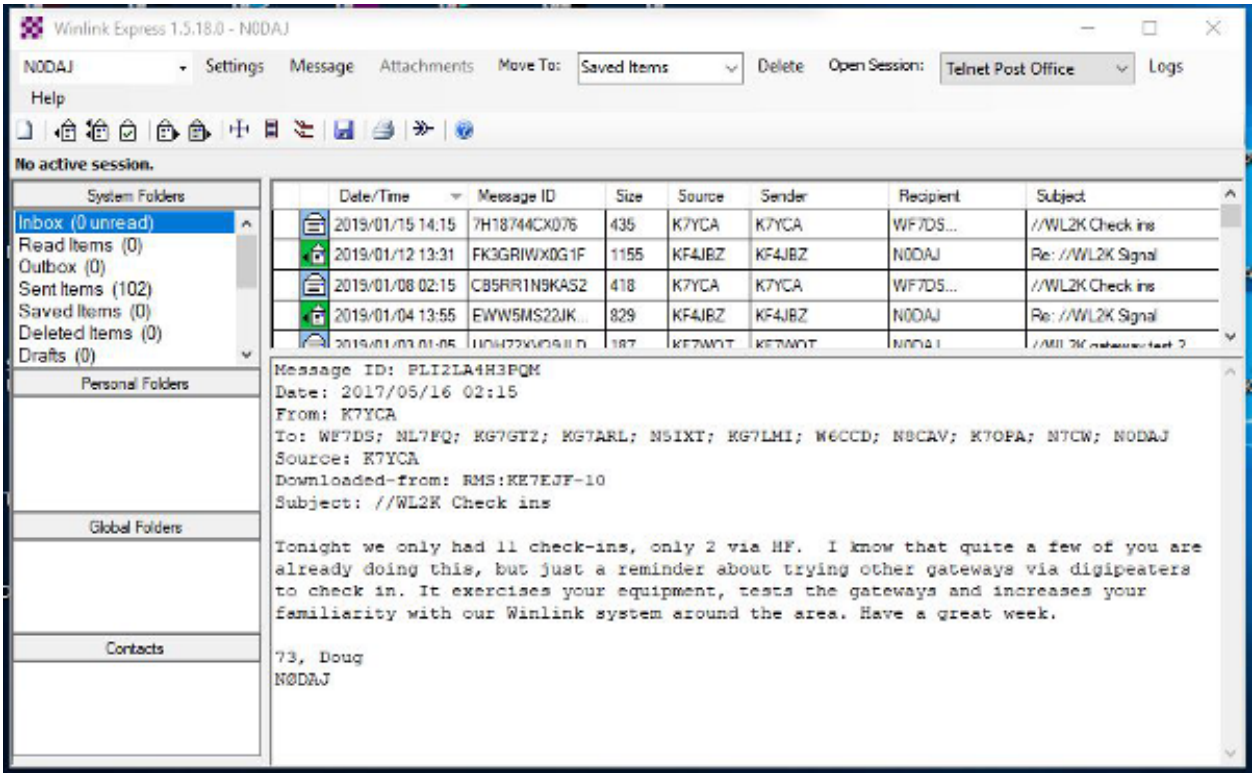
You will need to have detailed knowledge and skills when building your own open source email server, with the advantage of having complete control over everything on the system. There is some administrative overhead for creating and maintaining all user email accounts as well as handling other management 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.




## 16.3 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.




## 16.4 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 |

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# SEVENTEEN

# 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.

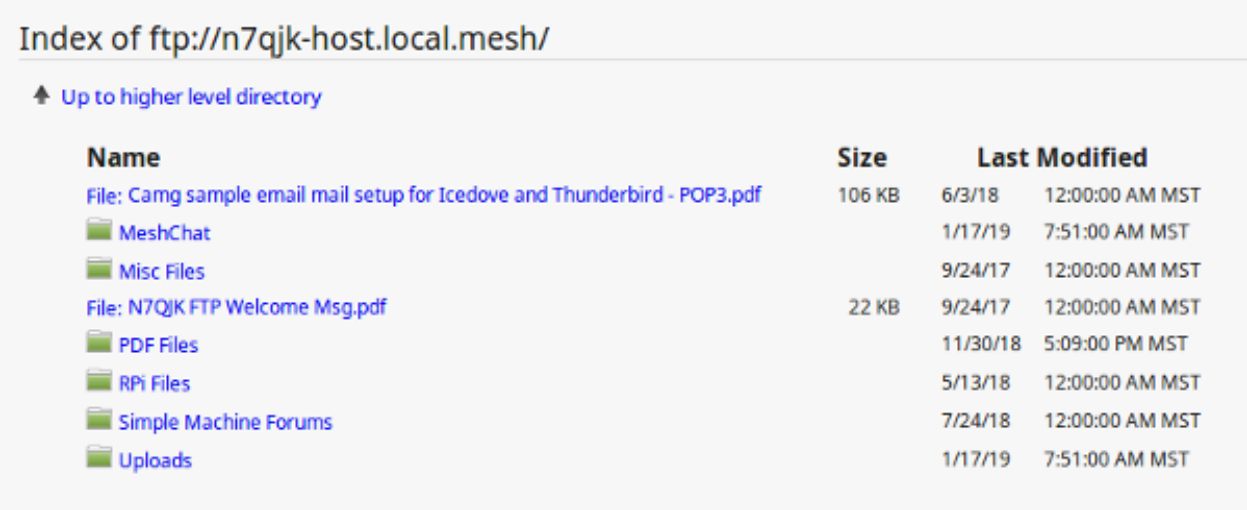
File transfer protocols themselves have minimal impact to network performance, but downloading a very large file across a mesh network could have a major performance impact. Transferring text files, and especially compressed text, should have minimal impact to the network, but a network could experience 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.

## 17.1 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.

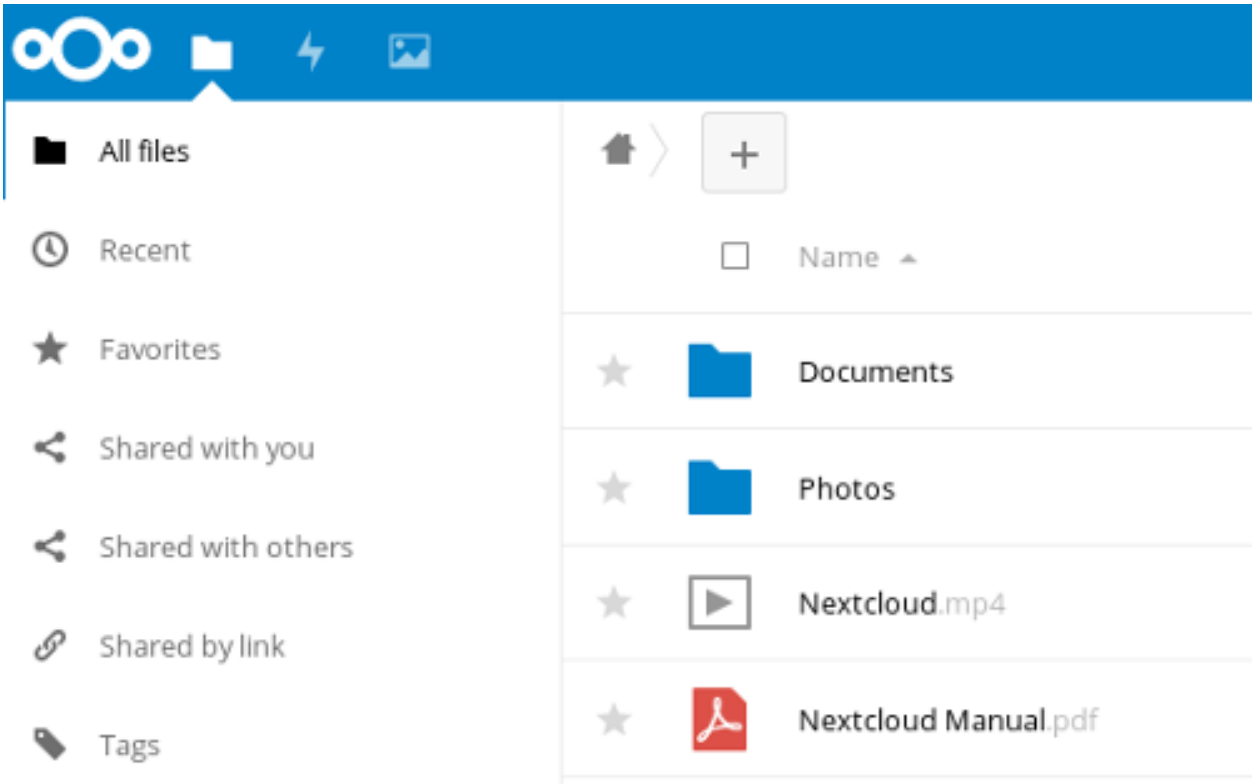



## 17.2 Web Services

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.



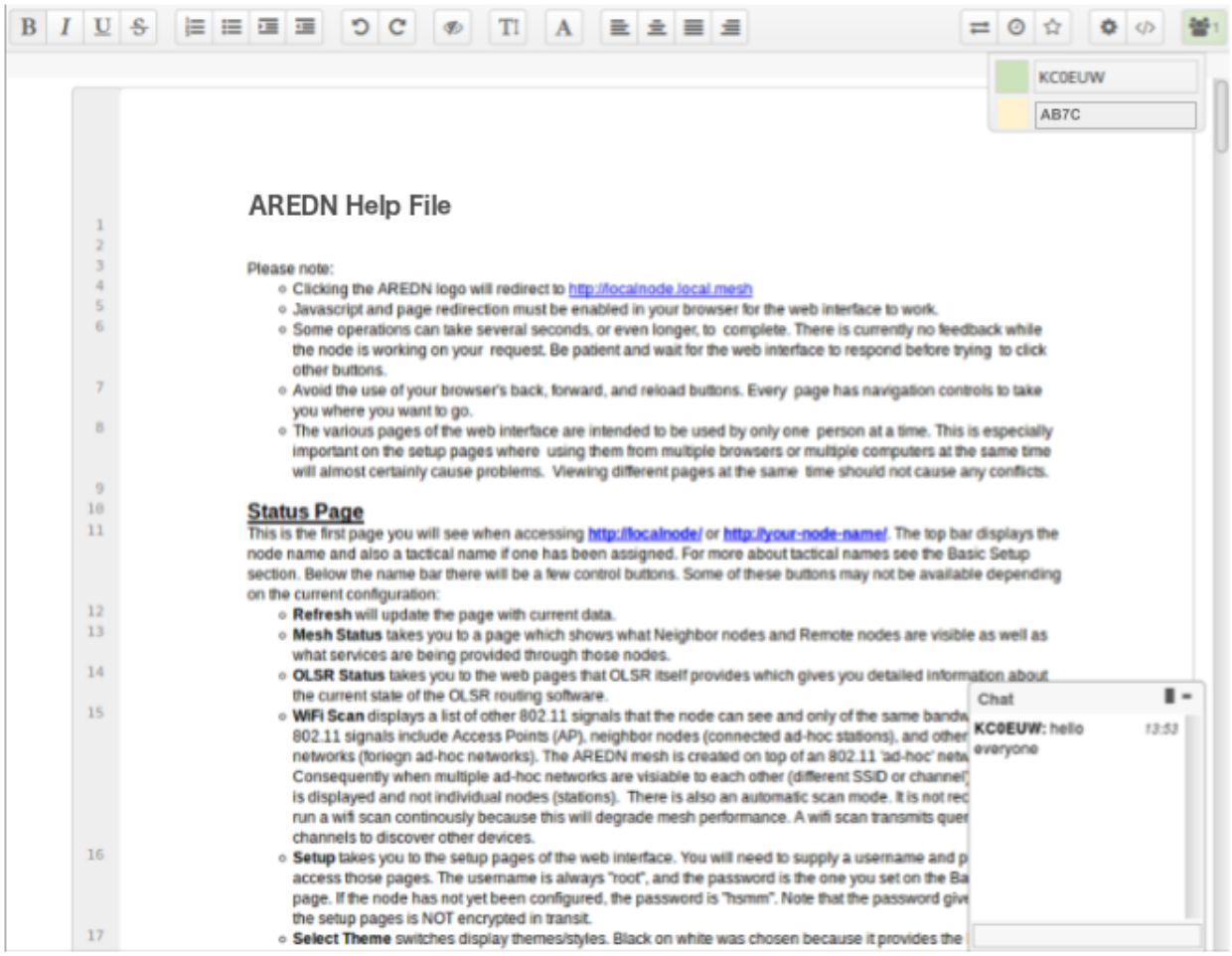

## 17.3 Collaborative Computing

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.




Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# EIGHTEEN

# 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.

## 18.1 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 pack-

ages. 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.

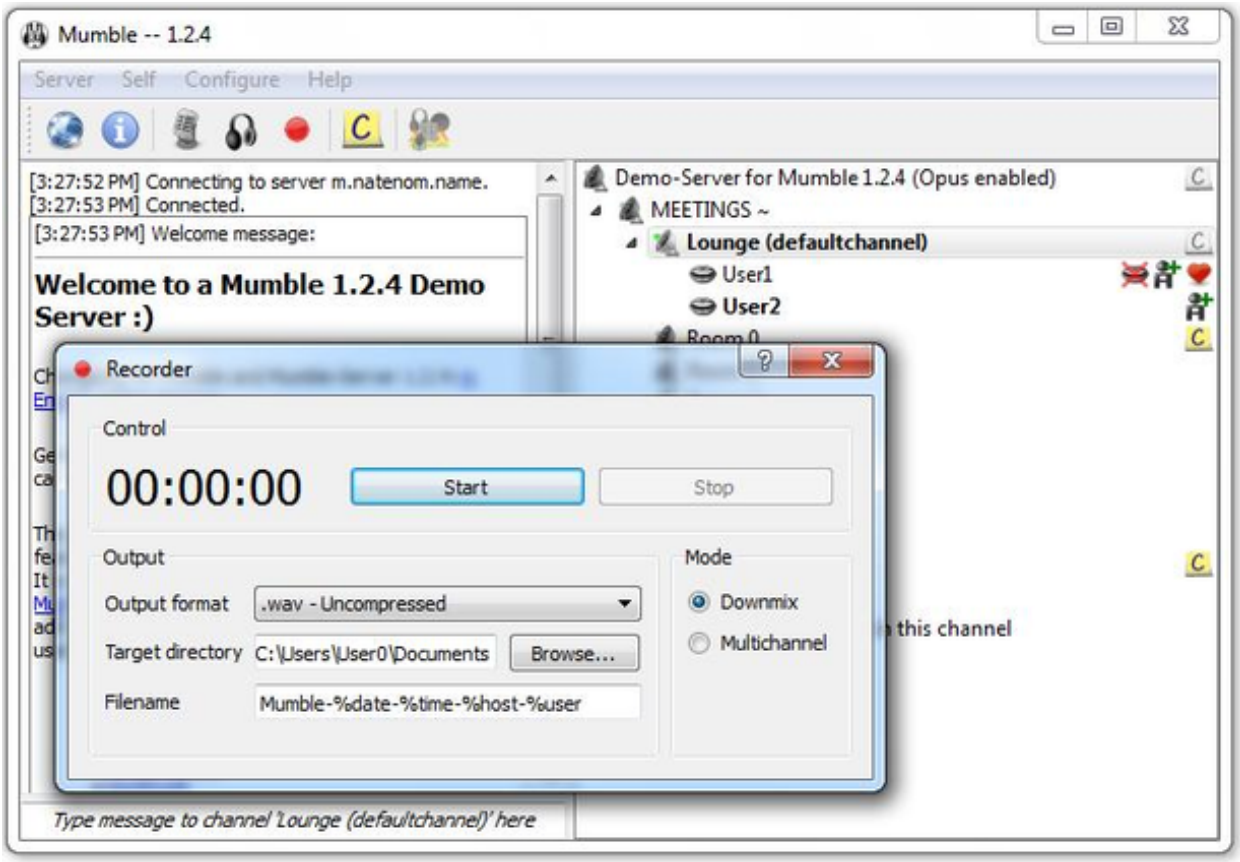

## 18.2 VoIP Endpoints

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.

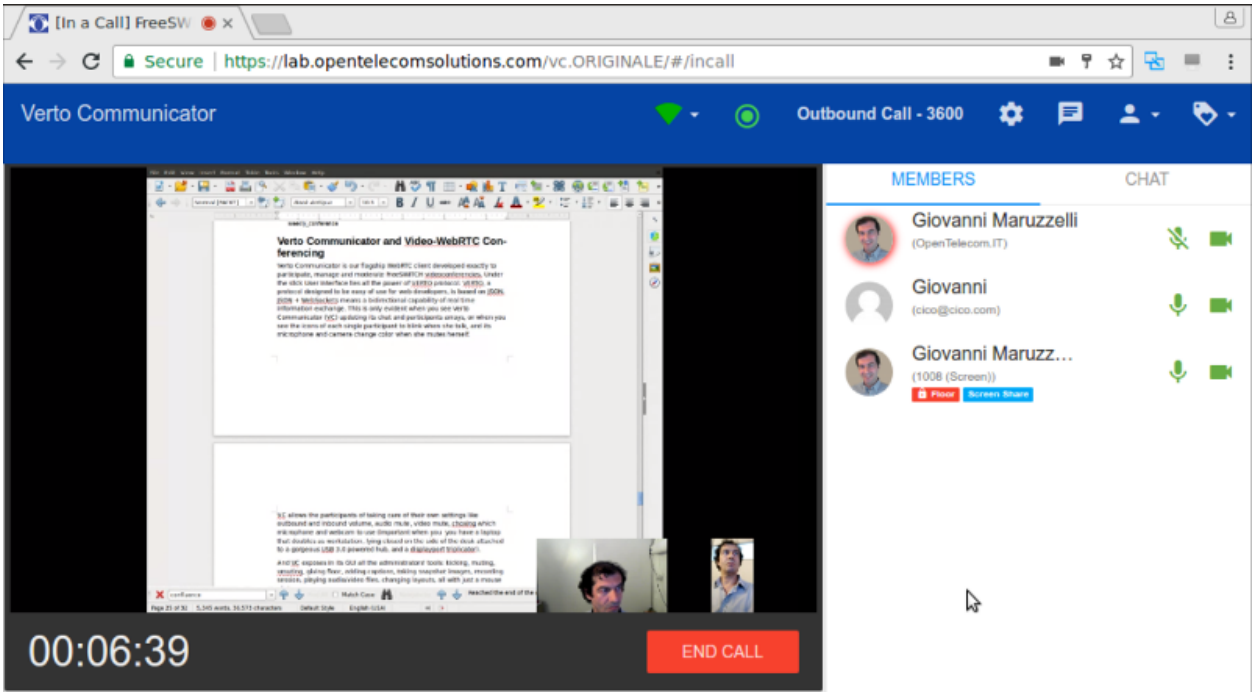
Hosting Mumble locally requires downloading the *Murmur* server, which is included as an option in the Mumble installer. The primary users of Mumble are Internet video gamers who want to communicate with each other during game play. However, it can also be used as a non-gaming voice communication service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




## 18.3 Video Conferencing Software

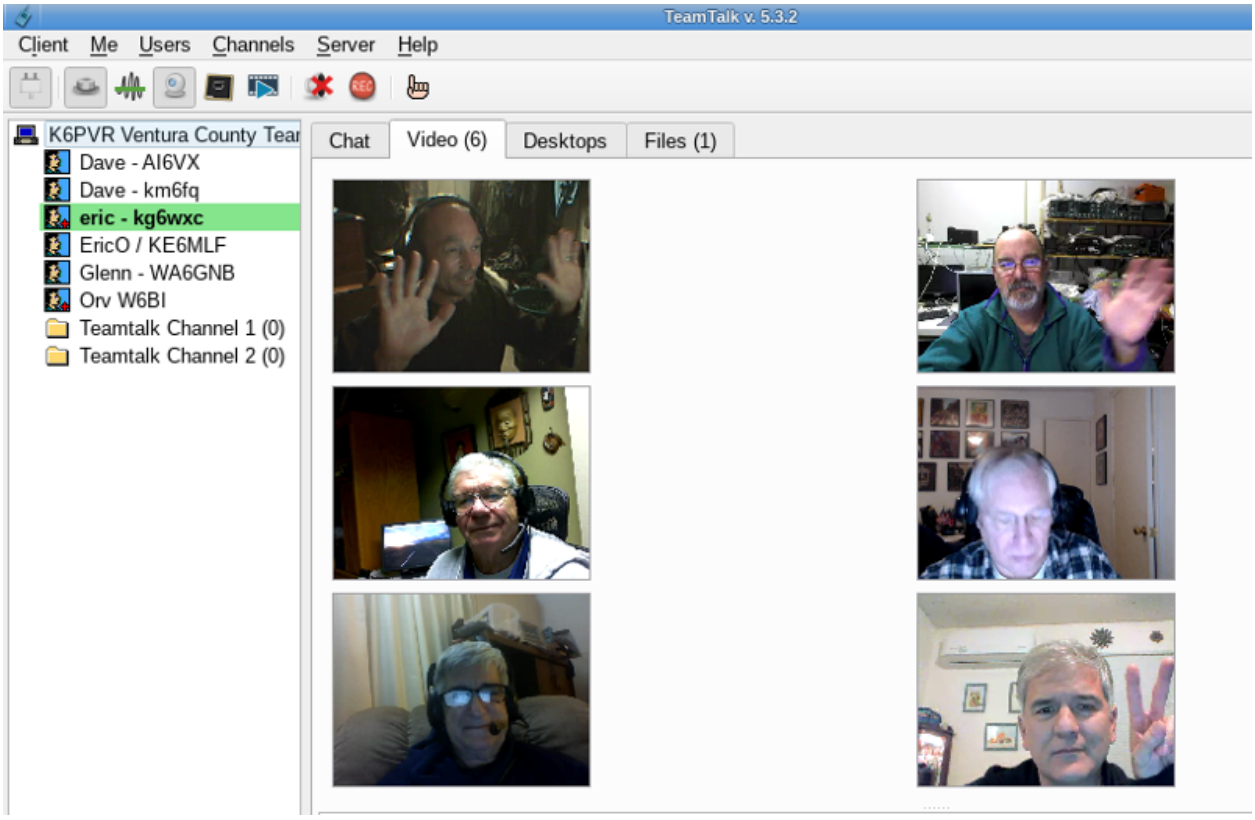
**FreeSWITCH Server** FreeSWITCH is a recent communication platform that can be used to build voice PBX systems with voice response menus, video conferencing with chat messaging and screen sharing capabilities, and full WebRTC support. Its modular design makes it possible to install only what is required 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.




## 18.4 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 |

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# NINETEEN

# 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."

## 19.1 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.

## 19.2 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.

### 19.2.1 iSpy

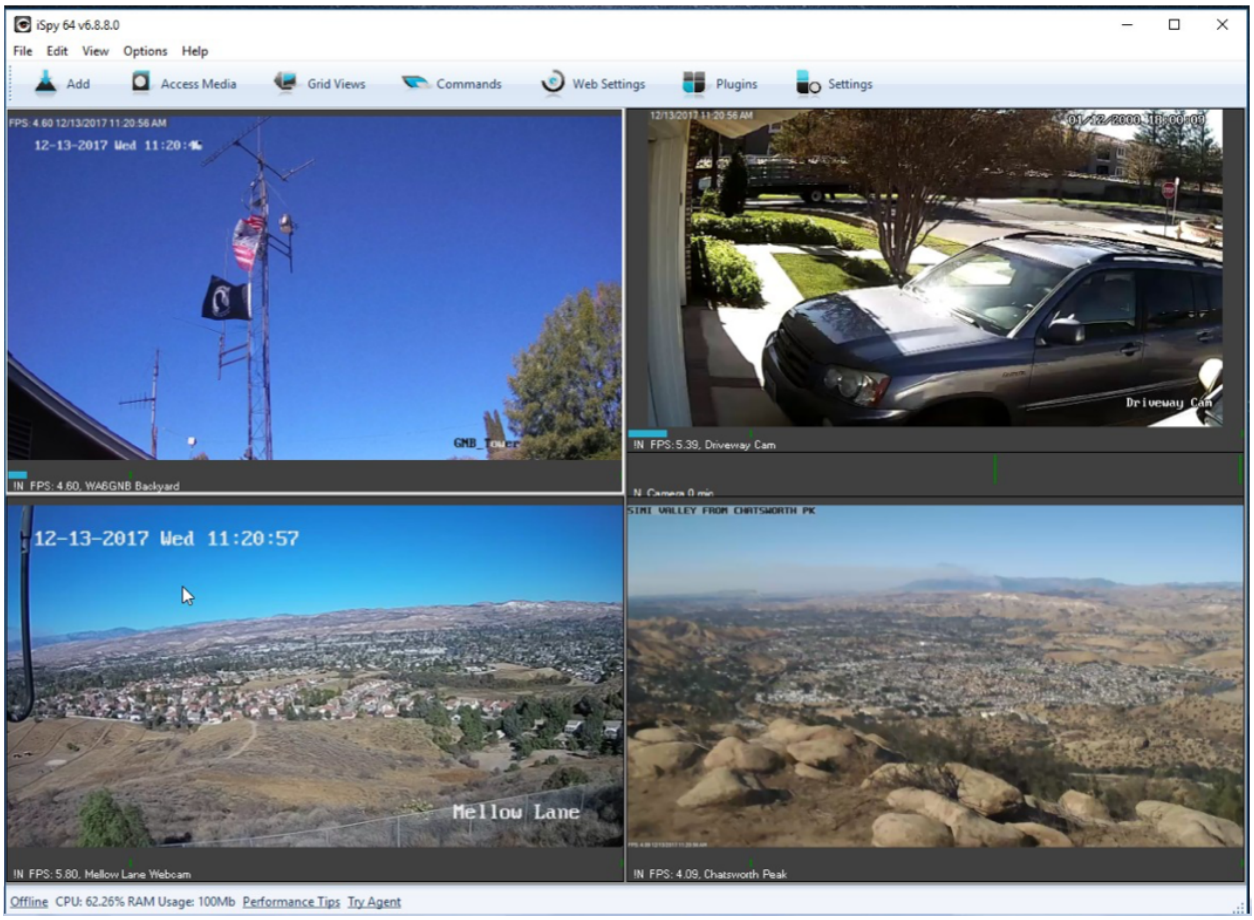
iSpy is a popular video management package for Microsoft Windows computers. It is certified on Windows 7 and above but may work on other systems that support the .NetV4 Framework. iSpy runs as a Windows program with a local user interface (UI) accessible on the computer on which it was installed. Additional 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`.

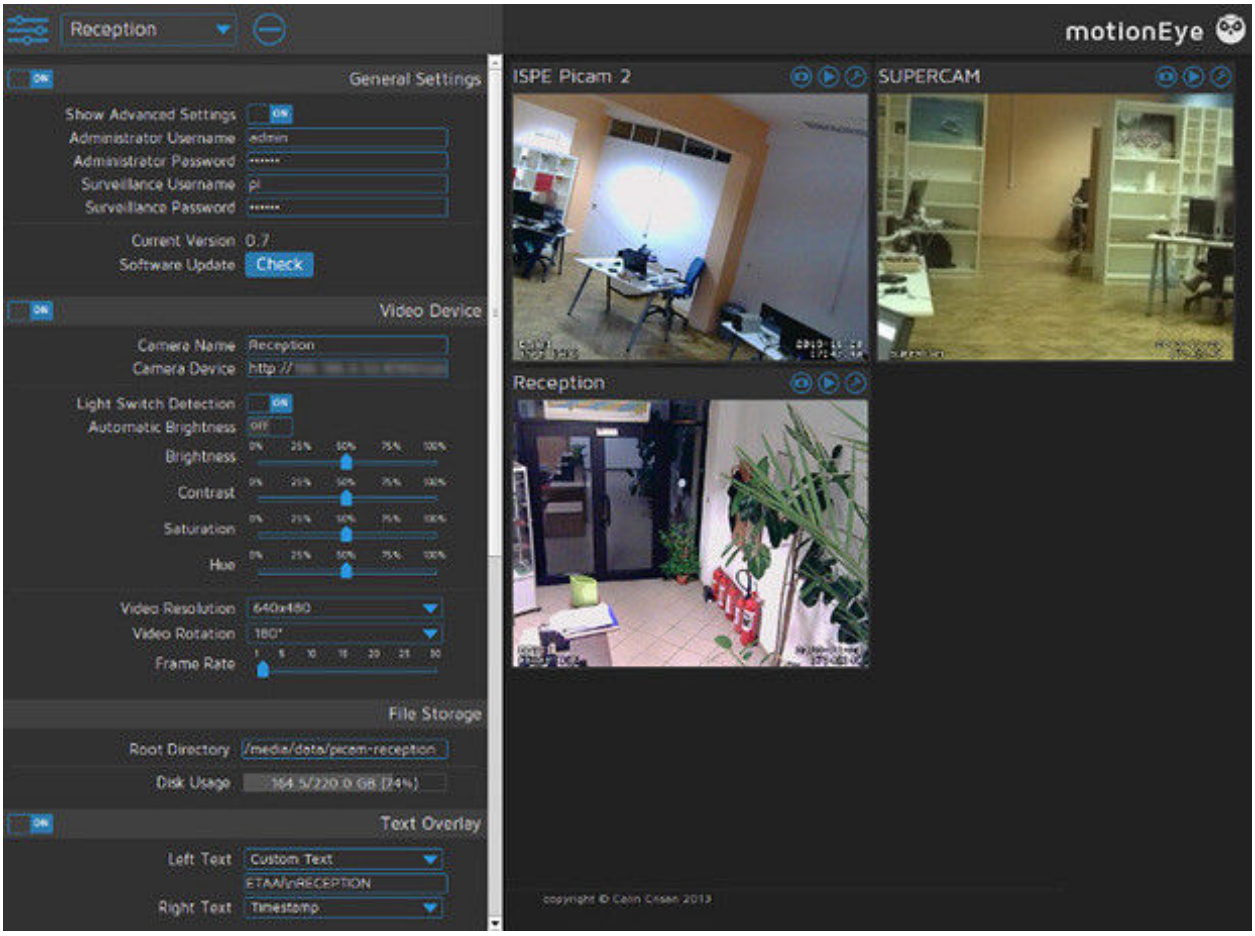
In the lower right video stream on the iSpy display below you can see the smoke plume from the 2017 Thomas Fire in California, which was recorded by a camera on the local AREDN® network. For additional information about iSpy, visit this link: iSpy.



### 19.2.2 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

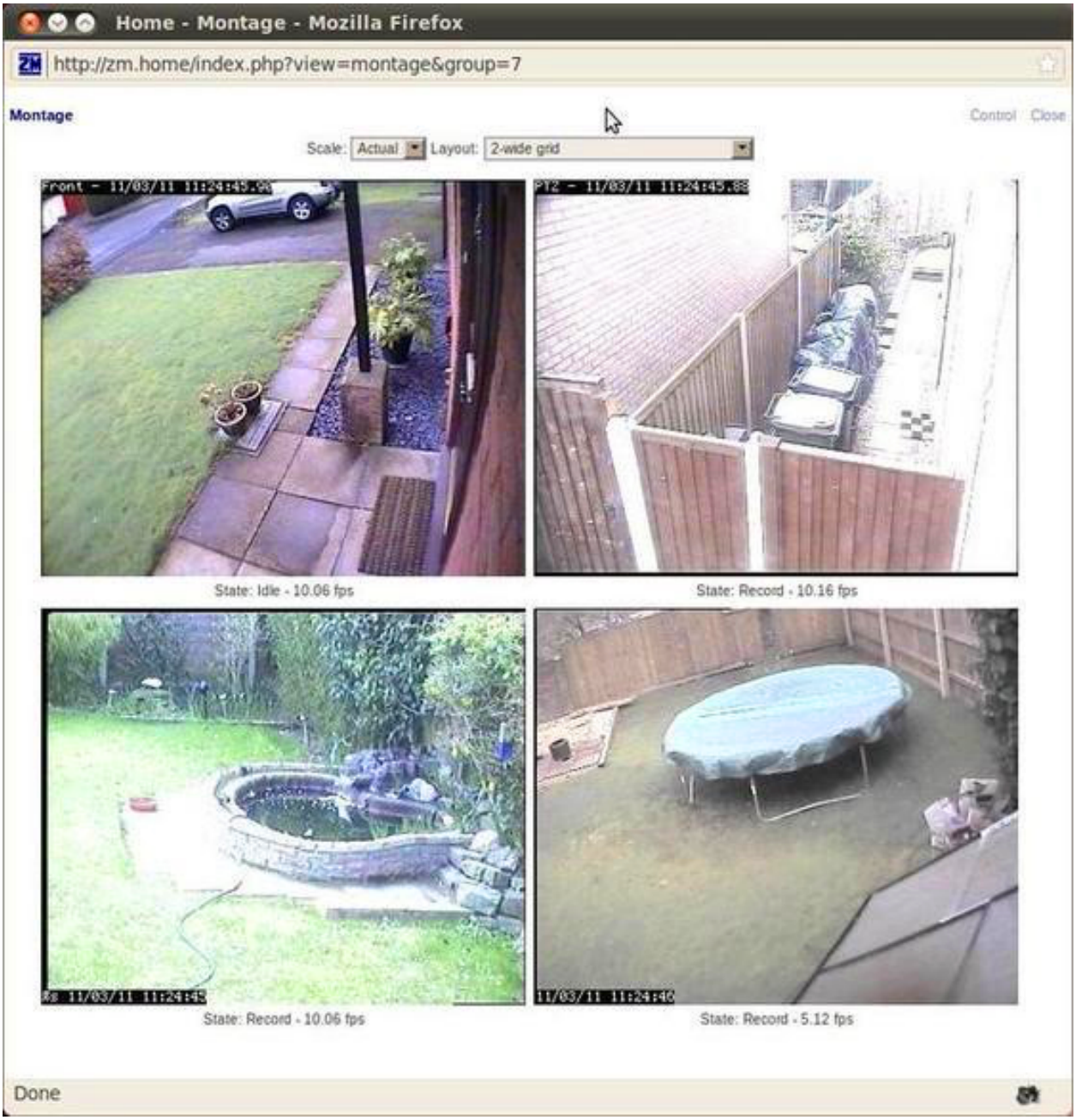


### 19.2.3 ZoneMinder

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.

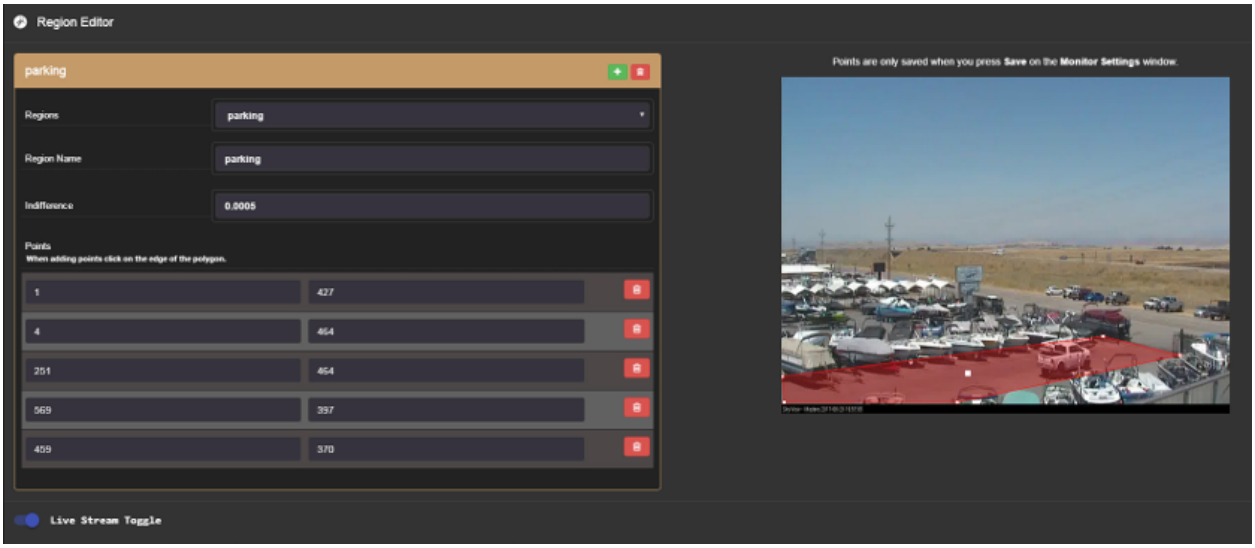

Home - Montage - Mozilla Firefox
http://zm.home/index.php?view=montage&group=7
Montage
Control
Close
Scale:
Actual
Layout:
2-wide grid
State: Idle - 10.06 fps
State: Record - 10.16 fps
State: Record - 10.06 fps
State: Record - 5.12 fps
Done

### 19.2.4 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.



## 19.3 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*

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

TWENTY

# 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.

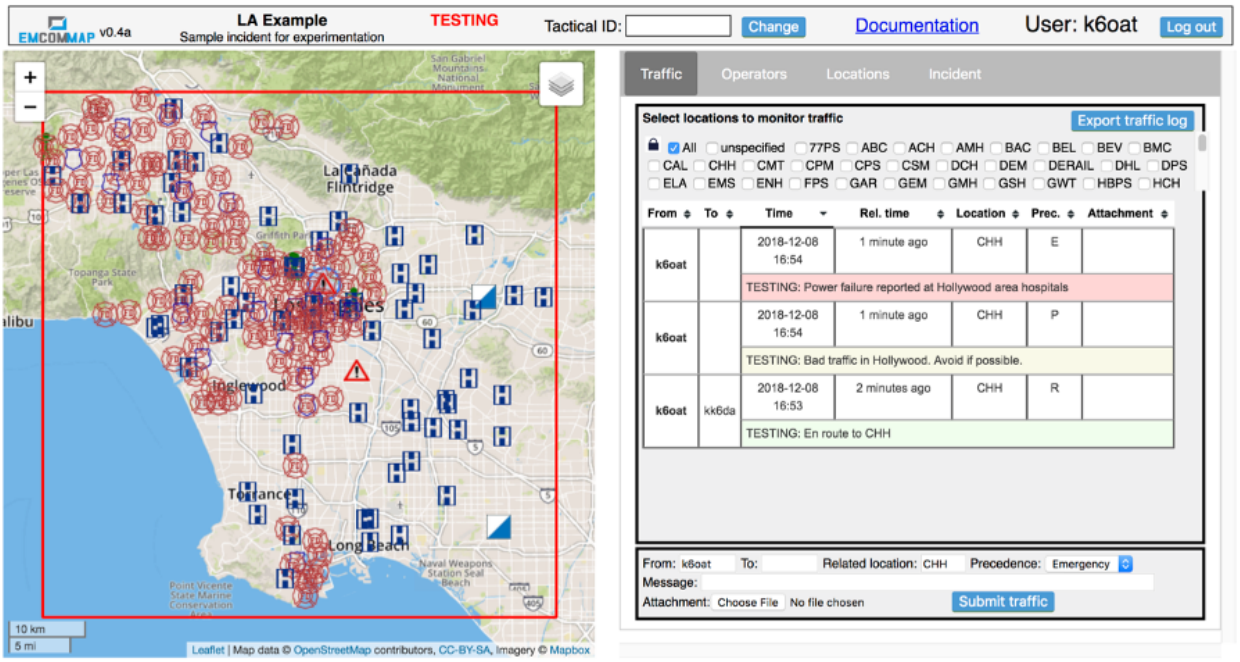
Served agencies have been using Computer Aided Dispatch (CAD) software for quite some time, and it has become their preferred method for managing events and incidents within their jurisdiction. In emergencies when electrical power or mission-critical facilities become unavailable and agencies are forced 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.

## 20.1 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.

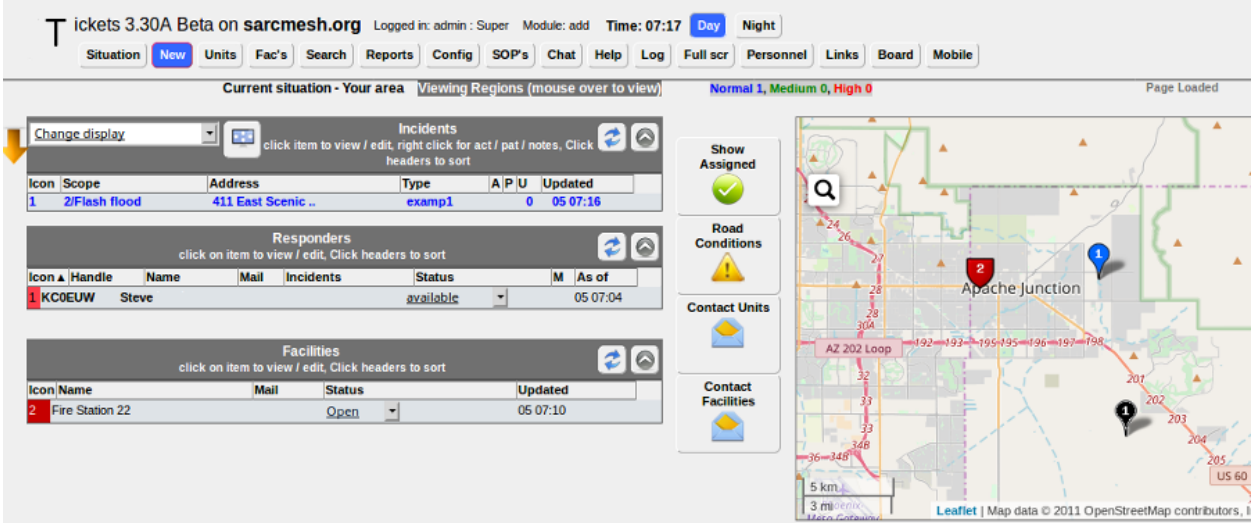
All communications are tracked and can be exported in spreadsheet format for offline use. Message traffic can be filtered to view specific messages for selected locations, and the traffic table can also be sorted for viewing the details based on information in any column. Message severity levels and 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.

## 20.2 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.




## 20.3 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 |

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

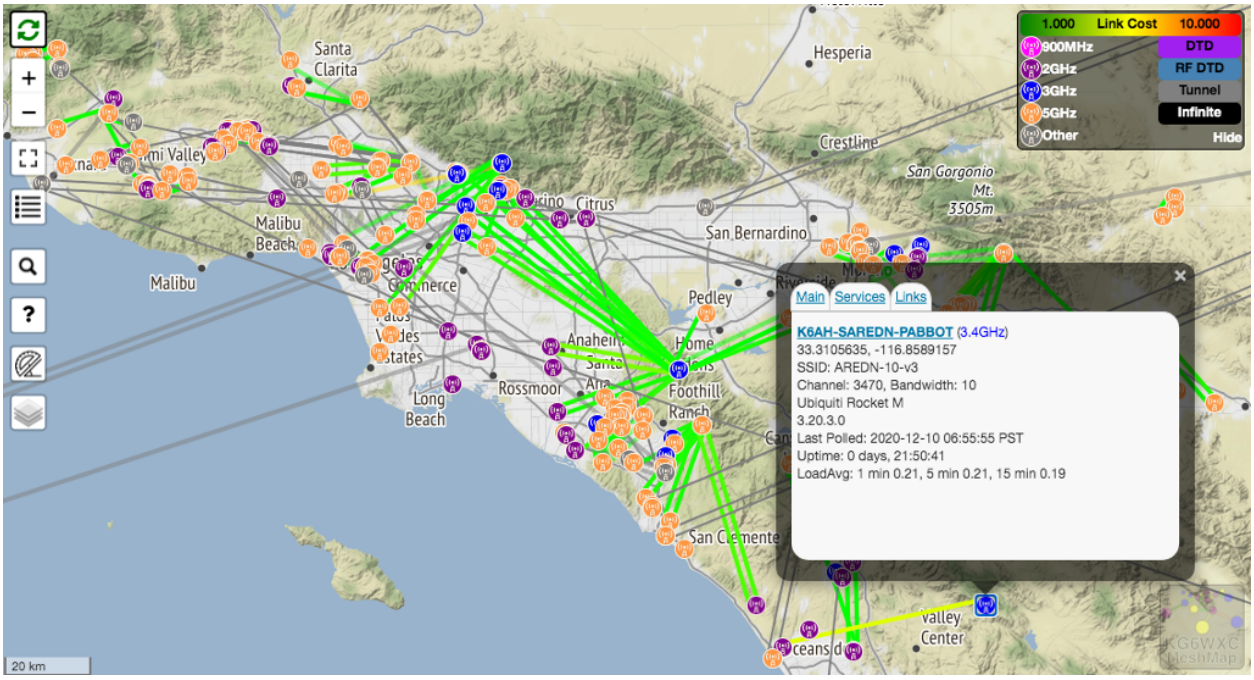
# TWENTYONE

# 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.

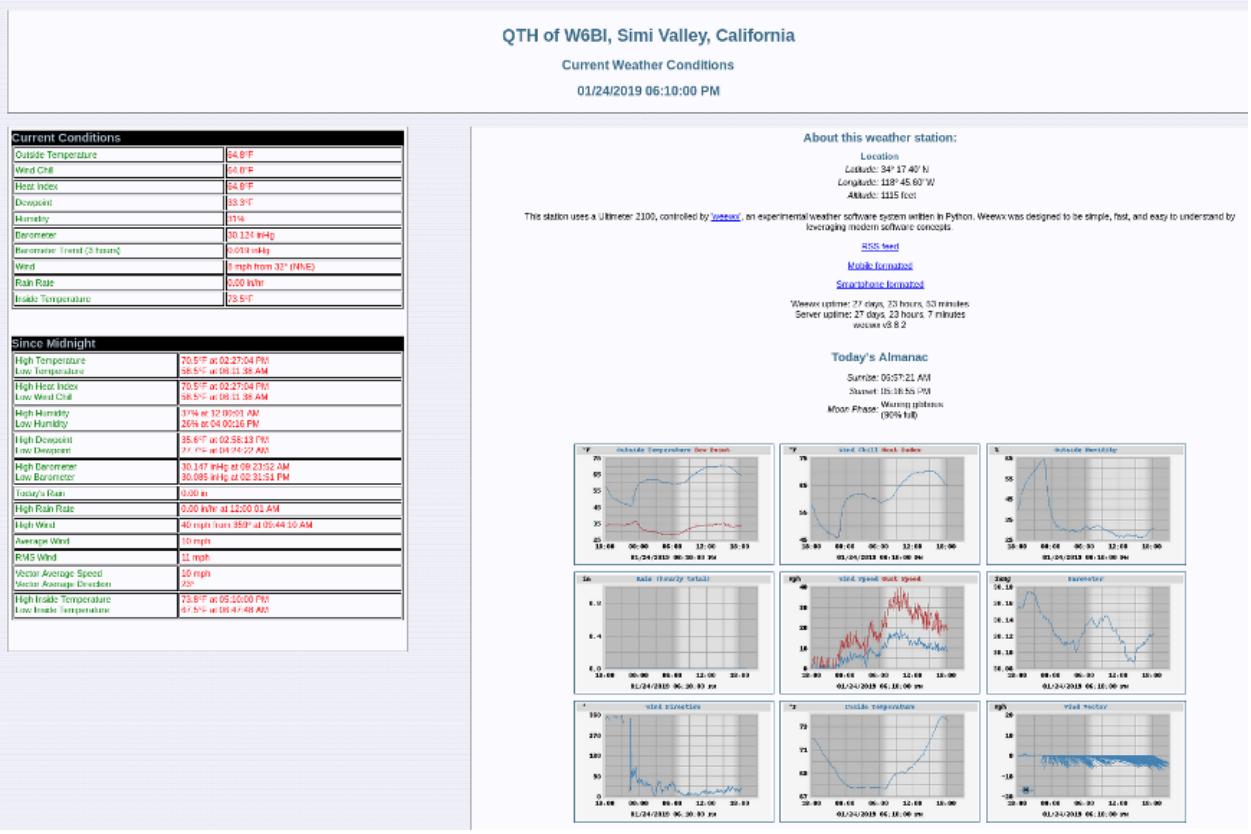
## 21.1 MeshMap Network Visualizer

**MeshMap** provides an automated way to visualize your AREDN® network topology. Eric KG6WXC created this useful tool and makes it available as an open source project. MeshMap can be installed on a mesh services computer having minimal hardware specifications, which allows it to run on a Raspberry Pi in your shack or in the field. MeshMap automatically discovers live nodes and periodically polls them to display their current configuration, services, and network link information. For additional information about MeshMap, visit this link: meshmap.



## 21.2 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.



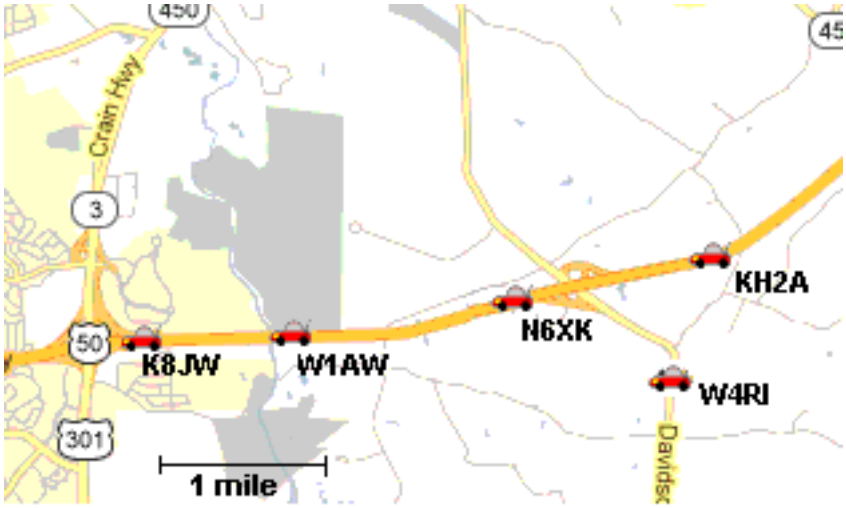
## 21.3 Network Time Services

Although the AREDN® nodes themselves do not depend on network time synchronization, there may be other programs or services running on your mesh 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. This may be especially helpful for devices that do not have an onboard realtime clock, such as Raspberry Pi computers. It may also be important to have accurate timestamps across the network for programs such as email message logging, file updates, video surveillance images, and many others.

Most NTP implementations depend on an Internet connection in order to synchronize with upstream time servers. However, it would be more useful to be able to synchronize system clocks in an off-grid situation when AREDN® nodes are deployed during an emergency. One way to accomplish this would be to configure one or more battery powered computers 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 can get a fix on as many GPS satellites as possible.


In order for NTP to operate properly, each client device must have a fast and reliable connection to the NTP servers on the network. Be sure to locate your NTP servers on reliable high-speed segments of your mesh. For additional information about building an off-grid NTP server, visit this link: G4WNC NTP.

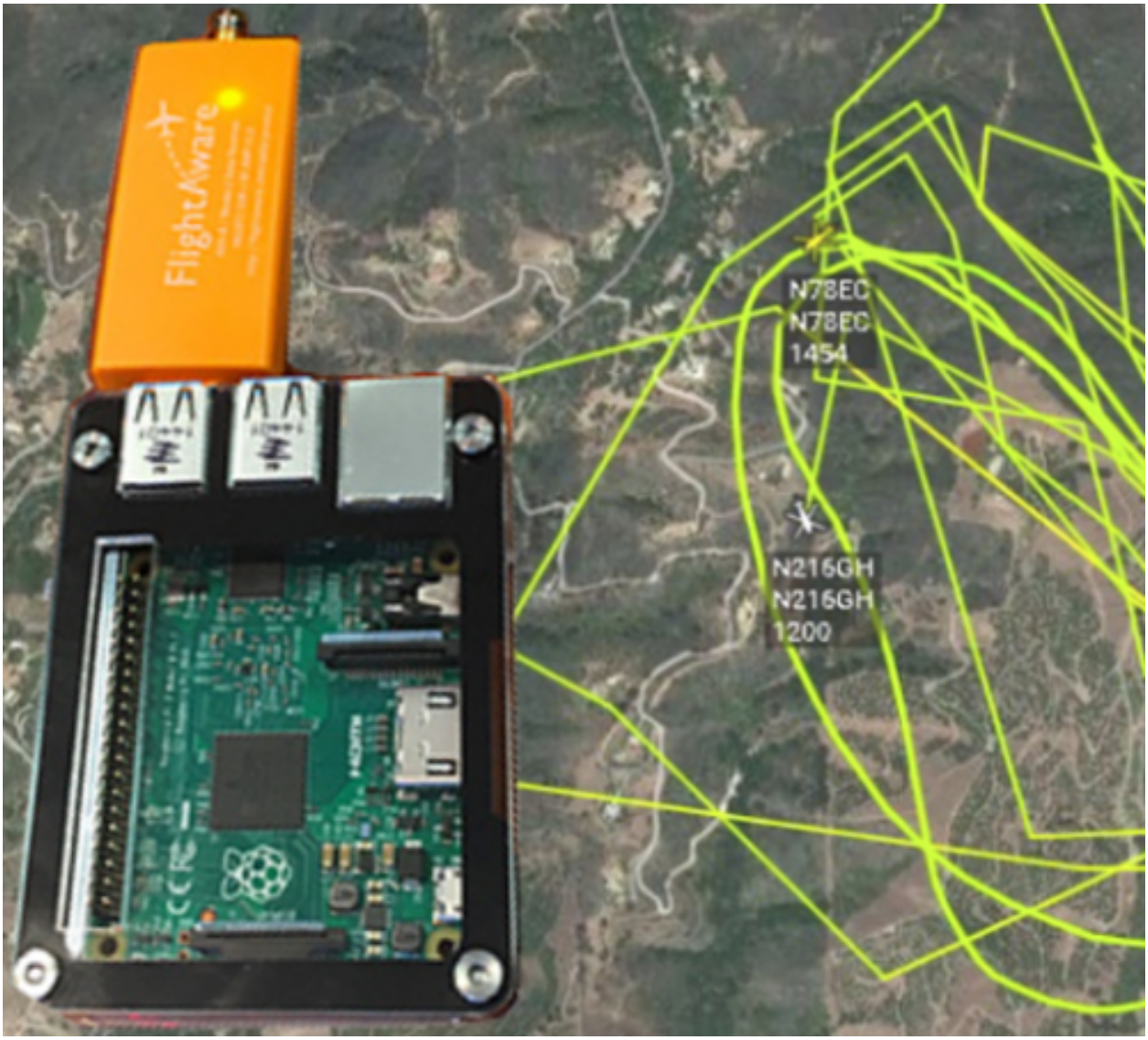
## 21.4 GPS Tracking Services




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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# TWENTYTWO

# TIPS FOR UPLOADING 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.

**Error message when uploading firmware** If you see an error message displayed when uploading new firmware to your node, verify that you are loading the correct file by referring to the AREDN download page, then you can safely ignore the warning. The file naming standard recently changed from a non-standard naming convention to the standard naming convention used by OpenWRT.

**Web browser cache and sessions** One common 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 setup and display an ERR_CONNECTION_RESET message. Clearing your computer's web browser cache will allow the latest URL settings to be registered so you can continue with the install process.

Instead of a *Connection Reset* message, sometimes a *Bad Gateway* message may appear. This is an HTTP Status Code that can mean any of several things. Often it indicates a network communication issue between a web browser and a web server. During AREDN® firmware installs you can usually resolve a *Bad Gateway* issue by doing 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.

## 22.1 Tips for Upgrading

Upgrading an AREDN® node is accomplished using the *Setup* > *Administration* > *Firmware Update* feature on the node's web interface. 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 can fail due to lack of memory, but such a failure will leave the node running its previous firmware version. The following tips help ensure that memory utilization is at a minimum on the node.

**Release node resources** Before starting the firmware upgrade on low memory devices, it may be necessary to stop, disable, or uninstall extra packages such as Meshchat, snmp, and tunneling. The goal of this step is to keep those processes from using RAM memory and to free as much RAM as possible before the upgrade. Rebooting the node before beginning the upgrade will ensure that RAM utilization is at a minimum.

You may also want to stop node programs or services that are not needed during the upgrade. For example, you can telnet or ssh to the node and type the command `wifi down` to free the memory used by this driver.

**Tips for legacy nodes with low memory (32mb)** Legacy equipment with only 32mb of memory may require more effort to upgrade. Be sure not to use these types of devices at sites which are difficult to access.

- You may need to try the sysupgrade procedure several times before it succeeds. Be patient and keep trying.
- Get everything ready to do the upgrade, then do a fresh reboot of the node and immediately start the sysupgrade process before the node has time to initialize services which use memory.
- Use command line access to copy the *sysupgrade.bin* image to the /tmp directory on the node, then run the sysupgrade process manually from the command line on the node. Note that AREDN® nodes use port 2222 for secure copy and secure shell access.

  Execute the following commands from a Linux computer:

```
>>>
my-computer:$ scp -P 2222 aredn-firmware-filename.bin root@192.
↪168.1.1:/tmp
my-computer:$ ssh -p 2222 root@192.168.1.1
~~~~~~~ after logging into the node with ssh ~~~~~~~
node:# sysupgrade /tmp/aredn-firmware-filename.bin
```

To transfer the image from a Windows computer you can use a *Secure Copy* program such as WinSCP. Then use a terminal program such as PuTTY to connect to the node via ssh or telnet in order to run the sysupgrade command shown as the last line above.

- As a last resort, use the TFTP procedure to load the *factory.bin* firmware image to the node. This procedure is described in the *First Install* sections of **Installing AREDN Firmware**.

Link: AREDN Webpage

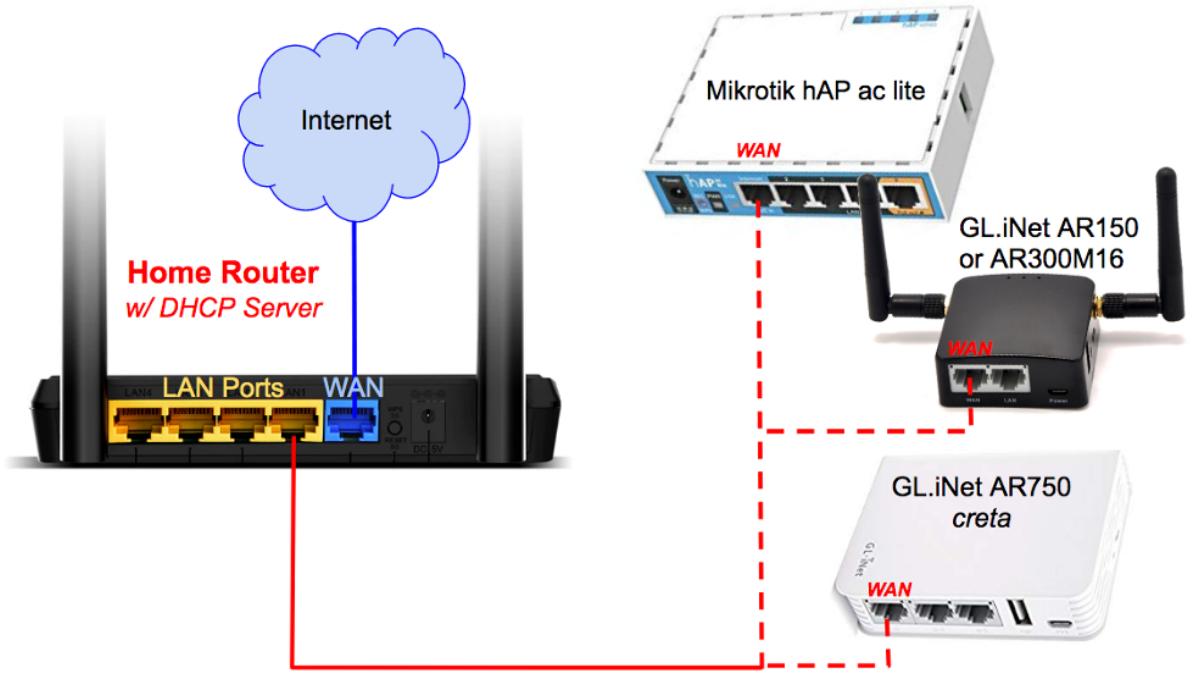
Link: AREDN Webpage

CHAPTER

# TWENTYTHREE

# CONNECTING NODES TO HOME ROUTERS

There are several indoor AREDN® nodes that have more than one Ethernet port, including the *Mikrotik hAP ac lite* as well as the *GL.iNet AR150, AR300M16*, and *AR750 Creta*. 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




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 other Mesh RF connected nodes, therefore be sure to leave *"Allow others to use my WAN"* unchecked. If you do not want any of your node's LAN connected devices to access the Internet either, you can check *"Prevent LAN devices from accessing WAN"*.

Link: AREDN Webpage

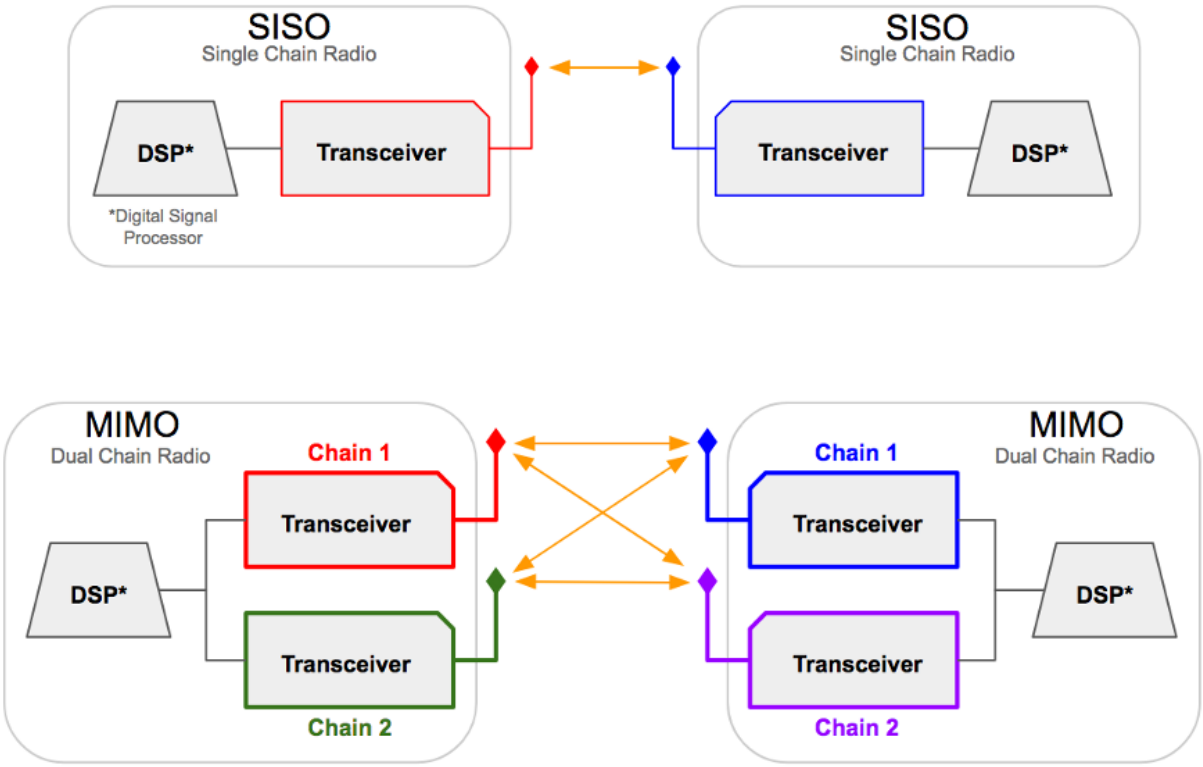
Link: AREDN Webpage

CHAPTER

# TWENTYFOUR

# 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 rate is always used. The selected MCS includes the appropriate modulation, forward error correction, and number of data streams.

## 24.1 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.

## 24.2 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.

## 24.3 SISO - MIMO Combinations

Today's mesh networks are likely to contain a mixture of single and multiple chain devices, so it is important to understand how different combinations of devices might perform.

**SISO to SISO** All transmit power is sent using a single polarization, but multipath signal combining does not occur. Only one data stream at a time can be sent at a rate that is limited by the protocol.


**SISO to MIMO** All transmit power is sent using a single polarization, and the MIMO receiver will enhance reception by combining multipath signals using MRC. Only one data stream at a time can be sent at a rate that is limited by the protocol.


**MIMO to SISO** The total transmit power is shared between MIMO chains, so the RF energy which is 90 degrees off-polarization from the receiving antenna may be lost. The SISO receiver cannot enhance multipath signals using MRC. Only one data stream at a time can be sent at a rate that is limited by the protocol.


**MIMO to MIMO** The total output power is shared between MIMO chains, but the full power from both polarizations can be processed by the receiver so that nothing is lost. The MIMO receiver can enhance reception by combining multipath signals using MRC. Simultaneous data streams can be sent using spatial multiplexing, effectively doubling data throughput.


## 24.4 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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# TWENTYFIVE

# 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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# TWENTYSIX

# 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.

## 26.1 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.

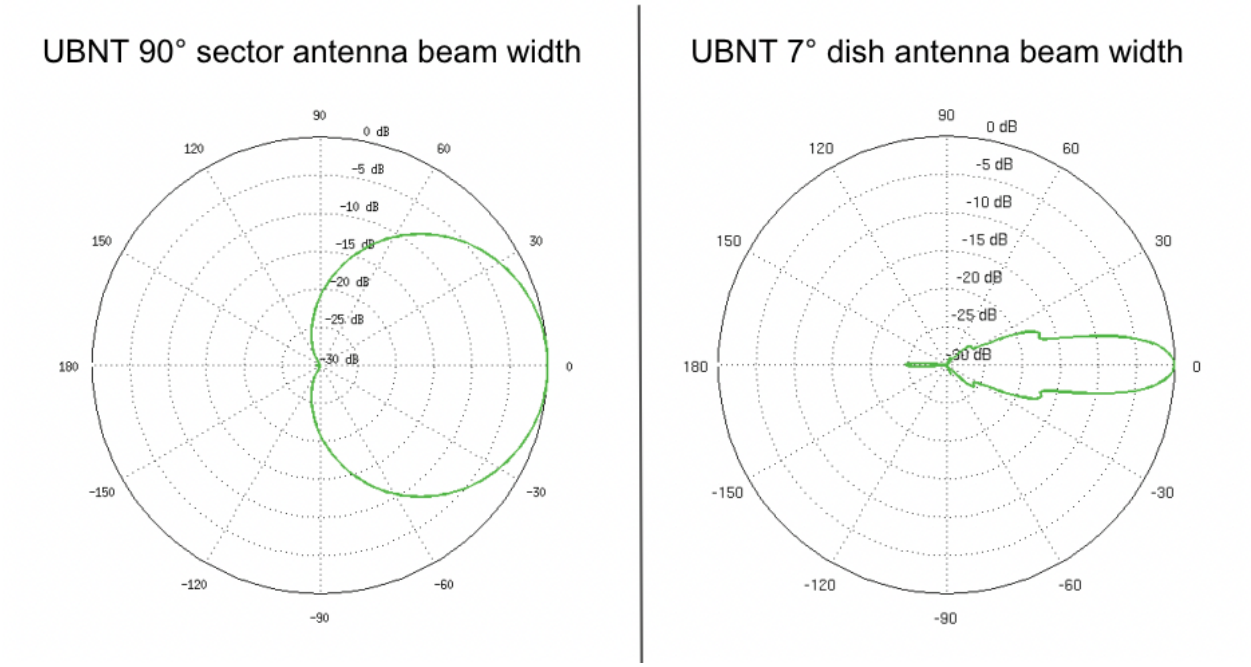
- In your vehicle, power up your node and plug in your laptop. Disable the wifi interface so the laptop gets its IP address from the node. Open a web browser and use *localnode.local.mesh:8080* to load your node's home page. You will need to have your user name (root) and password to authenticate to the *Setup* display.
- Enter the SSID, Channel, and Channel Width that matches the remote node you are surveying. Regarding the "Distance to Farthest Neighbor" setting, refer to the node help page or the *Configuration Deep Dive > Mesh RF Column > Distance Setting* section in the **Getting Started Guide** for information. On short paths the zero-distance (automatic setting) may not work well, so you should adjust the slider to a setting close to the estimated distance between your nodes. If you changed any of these settings, click `Save Changes` followed by `Reboot`.
- Now you can do a `WiFi Scan` from your node's home page. Put the scan on `Auto` refresh and the screen will refresh the scan every ten seconds. The scan list will show remote nodes along with their signal strength, channel number, and SSID. If you have chosen the correct SSID and channel, you should see a connected status if the signal is -87 or stronger. If the channel or SSID doesn't match, you will see a "foreign network" status. There may be other devices on different channels at a particular location. Pick the strongest one and use that channel.

- Once you have a connection with the remote node, quit the WiFi scan and click the `Charts` button. You will see a moving graph for the average of all connected stations. In the dropdown menu, choose the remote node you are connected to. Click the *Sound:* `On` button, and the pitch of the tone you hear will get higher with greater Signal-to-Noise Ratio (SNR). You may want to adjust the level of the starting tone as well as the tone volume using the sliders below the sound button. You will see the SNR updated every second above the sound button.
- To get the highest tone pitch and the best SNR, turn your radio slowly or even change the car position by driving forward or back a few feet. If the tone stays at one frequency and the chart is no longer changing, you may have lost the signal. Quit the chart and start again.
- Once you have the highest SNR at your test location, quit the chart and click the `Mesh Status` button. You should see the remote node in the list of *Current Neighbors* on the right. There will also be percent values for LQ based on the signal your node hears, as well as NLQ based on the signal the remote node hears. Right-click the neighbor node link and open it in a new tab on your browser. In the new tab you can see the remote node's view of your connection.
- On the remote node's home page, click the `Chart` button and follow the same procedure as in the step above. This time choose your own node from the dropdown menu, since that remote node may be connected to other stations too. You can turn on the audio tone if you want to hear the relative strength of your node's signal from the perspective of the remote node.
- Now you can turn your node's antenna a little at a time in order to get the highest possible SNR that's being received by the remote node. You will probably notice less variation in the chart with small movements making it easier to adjust for strongest SNR.
- Quit the chart once you have the best signal level. If you hover your mouse over the chart you can also view the individual data points that show the specific transmit and receive signal levels (dBm). Check both your *Mesh Status* page and the neighbor's *Mesh Status* page for the LQ and NLQ values. Try to achieve 100/100 percent on each side.

## 26.2 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.

Another invaluable tool mentioned in the **Applications and Services Guide** under *Other Services* is KG6WXC's MeshMap Network Visualizer. This program automatically discovers live nodes on a mesh network and periodically polls them to display their location, configuration, services, and link information. It also has a ruler tool that displays the distance and true bearing (not magnetic) between any two points you select on the map.

Studying the types of maps mentioned above 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 easier 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.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# TWENTYSEVEN

# TEST NETWORK LINKS WITH IPERF3

iperf3 is an open source network throughput testing tool which is now included in the AREDN® firmware by default. It is a client-server utility, so it must be available on both nodes that participate in the test scenario. The iperf3 client node generates traffic which is sent to the server node. Network throughput is measured and an estimate of the network speeds between that client and server is displayed.

Understand the impact to your network before using iperf3. During the test period iperf3 will generate a significant amount of traffic in order to determine the capacity of the link between the client and server nodes. Try to run your iperf3 testing during times when you know that there will be minimal impact to the routine traffic between the nodes.

One of the many uses for iperf3 is to validate and optimize your node's *Distance* setting on the **Basic Setup** page. Try different *Distance* settings and note the network throughput using iperf3, with the goal of choosing a *Distance* setting which yields the best network performance.

## 27.1 Using the Onboard iperf URL Feature

There is a simple, lightweight CGI interface that can be used to run an iperf3 test between two nodes which have firmware with this feature. From any computer connected to the network you can open a new web browser tab or window and type an iperf testing URL having the following format.

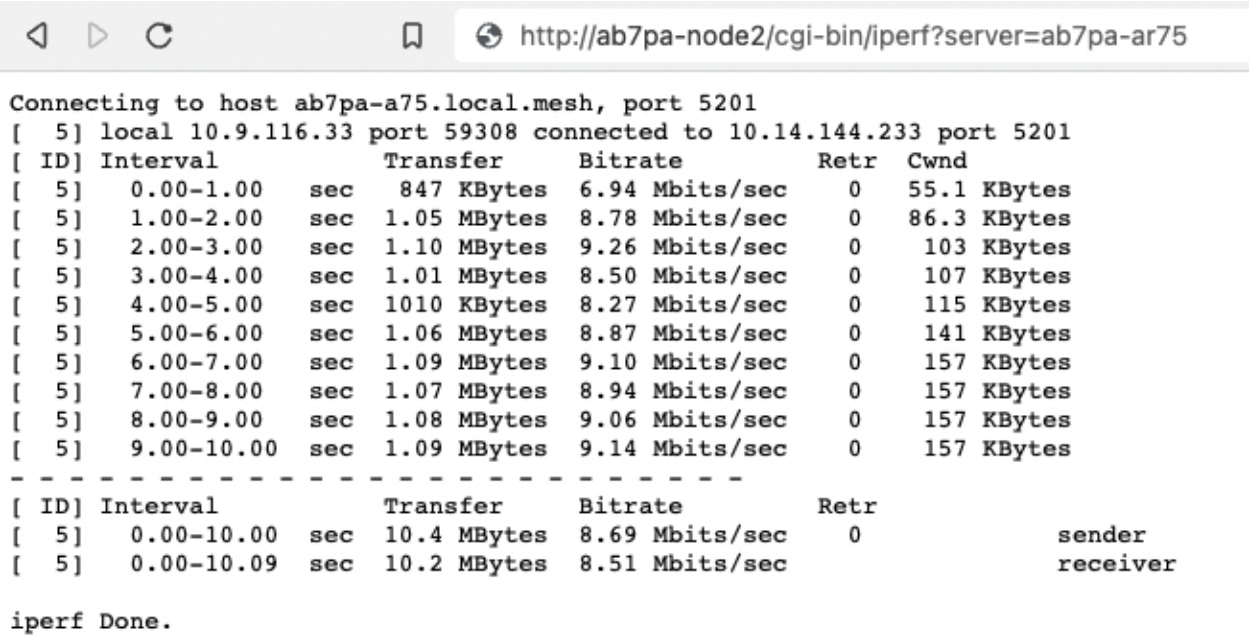
`http://<client_node_name>/cgi-bin/iperf?server=<server_node_name>&protocol=<tcp|udp>`

*Client Node Name* is the fully qualified node name for the client/sender node. If you do not include the "local.mesh" suffix then it will be added for you.

*Server Node Name* is the fully qualified node name for the server/receiver node. If you do not include the "local.mesh" suffix then it will be added for you.

The *Protocol* parameter is optional. If no protocol is specified, then a TCP test will be started. If you want to eliminate the typical TCP handshaking overhead on your network then you can run a connectionless UDP test by adding `&protocol=udp` after the server parameter.

Once you activate the URL in your web browser an iperf3 server will be started on the node you selected as the server, and the client node will initiate the iperf3 test using the protocol you specified (if any). Once the test has completed you will see the collected data summarized by time interval, and at the bottom of the display is the overall average from the perspective of the sender (client) and the receiver (server).

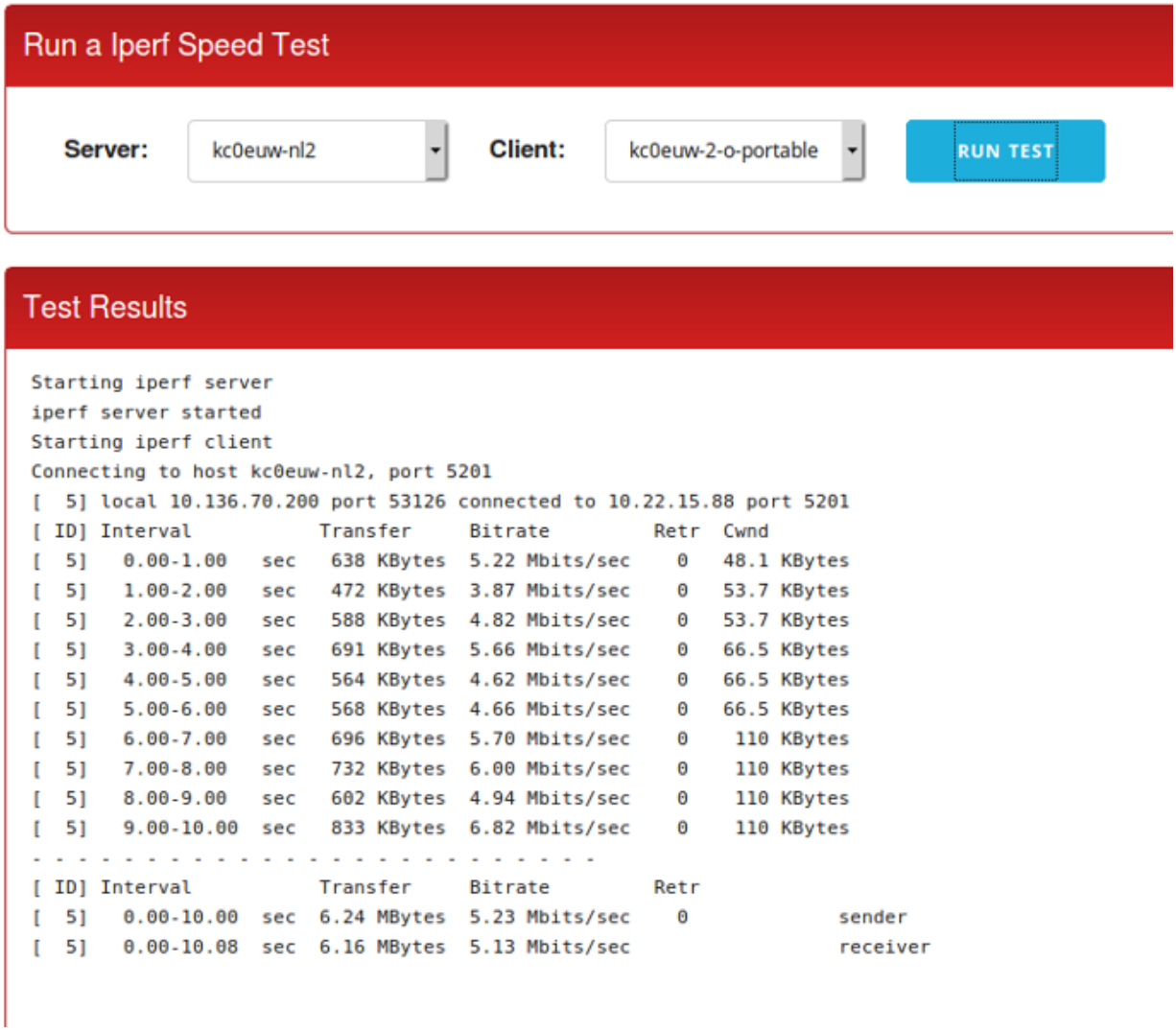



## 27.2 Installing and Using IperfSpeed

The **IperfSpeed** package provides a web-based control interface for running network tests between nodes, and it was written by Trevor Paskett K7FPV using the Perl programming language. With the project to retire Perl on AREDN® nodes, there is now an alternative *IperfSpeed* package which uses the Lua programming language. The original Perl and new Lua packages are available at the following links:

- Original Perl version of IperfSpeed
- New Lua version of IperfSpeed

Select the *IperfSpeed* service on one of the nodes to open its web interface in a new browser tab or window. From the dropdown lists, select a node as the iperf3 server and also one as the iperf3 client. Click the *Run Test* button to begin the network throughput test.




Once the test has completed you will see the results displayed in the *IperfSpeed* interface. *IperfSpeed* also tracks previous tests that have been run, and it allows you to rerun any of the previous tests by clicking the *Re-Test* button.

Link: AREDN Webpage

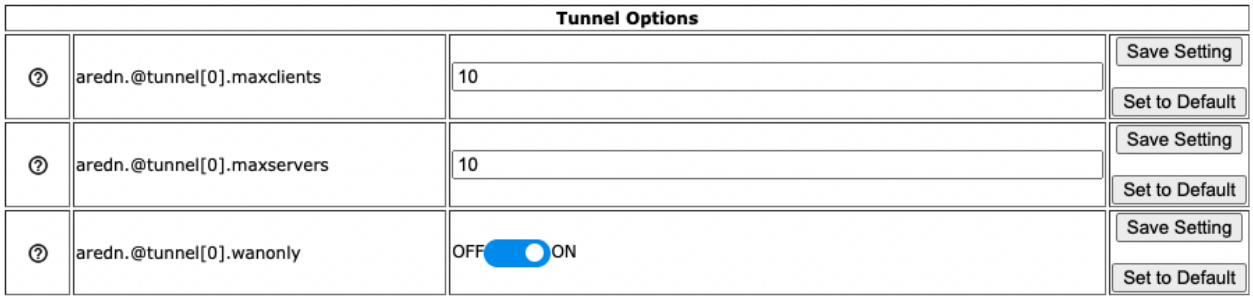
Link: AREDN Webpage

CHAPTER

# TWENTYEIGHT

# CHANGING TUNNEL MAX SETTINGS

By default a node is allowed to host up to 10 clients in its *Tunnel Server* display and connect with up to 10 servers in its *Tunnel Client* display. The *maxclients* and *maxservers* values on the **Advanced Configuration** page provide a method for adjusting the default settings.




**Warning:** Use caution when increasing the *maxclients* or *maxservers* values. Enter only *zero* or positive integers up to a maximum value for the number of active connections your node hardware can handle, since each active tunnel connection consumes system resources that the node may need for normal operation.

A node's CPU and memory utilization will increase with every additional tunnel connection. If you have node hardware with a faster CPU and more total memory, you may be able to host more clients or connect with more servers than on a node with a slower CPU and less total memory. Tuning these settings to their optimum value for your specific node hardware may involve some experimentation, as described below.

**Free Memory:** Each active connection consumes almost 1500 KB of memory on your node. You can see the current amount of free memory on the *Node Status* page, but actual memory utilization fluctuates constantly. The less free memory a node has available, the greater the chance that normal operational events could cause some processes

to fail unpredictably. Monitor the node's *Free Memory* over a period of time to estimate the resources required during normal operation. Then be sure to allow some memory headroom to account for occasional bursts of high demand. Once you understand how much memory your node typically requires (min/avg/max), and you know how much free memory headroom you want to reserve, you could divide the remaining free memory by 1500 KB to estimate the maximum number of active tunnels your node might support without suffering unpredictable failures.

**CPU Utilization:** Each active tunnel also consumes some fraction of the CPU, depending on the instantaneous amount of traffic in the tunnel. Activities such as *Meshchat* and VoIP phone calls generate less traffic than video streams and large file transfers. Even when there is no user traffic, mesh protocols themselves create a small continuous load that grows with the size of your network. If you use *telnet* or *ssh* to log into your node, you can run the *top* program from the command line to watch CPU utilization. Observe the effect as you transfer a large file, visit a complex web page, or view a video stream. Entirely saturating the CPU is unlikely to cause the node to fail, but it may cause other unexpected behavior. As with free memory, be sure to plan for a reasonable amount of idle CPU headroom.

If the *maxclients* or *maxservers* values are increased, then the *Add* row will allow more clients or servers to be added. If the values are decreased then the *Add* row will disappear when you have an existing number of rows greater than or equal to the new maximum value. Existing rows beyond the maximum value will still be displayed in order to retain any previously assigned credentials. If more credentials are *Enabled* than are allowed, then only the maximum number of tunnels allowed will be activated, beginning with the first *Enabled* row in the table. A warning message will be displayed if the number of *Enabled* tunnels exceeds the current value of *maxclients* or *maxservers*. While this warning is displayed, the system will not allow configuration changes to be saved. The warning can be cleared either by increasing the limit or by unchecking enough *Enabled* boxes to reduce the number of enabled tunnels to the limit.

These settings are runtime values, which means that if you change them you will see the effect immediately in the *Tunnel Server* and *Tunnel Client* displays without the need to reboot your node. If you adjust the *maxclients* or *maxservers* settings to values that change the number of allowable active connections, then when you navigate to the Tunnel Server or Tunnel Client pages you can click the *Save Changes* button in order to see the effect of your change on the existing active connections.

If you set the *maxclients* or *maxservers* value to zero, then the corresponding Tunnel Server or Tunnel Client display will not allow any rows to be added. This may be useful if you want to provide a tunnel server but you also want to remove the ability to connect with other servers as a tunnel client. The same is true if you want to connect to other tunnel server nodes as a client but you do not want provide any tunnel server capabilities on your node.

As stated above, use caution when increasing the *maxclients* or *maxservers* values. Enter only *zero* or positive integers up to a maximum value for the number of active connections your node hardware can handle, since each active tunnel connection consumes system resources that the node may need for normal operation.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

TWENTYNINE

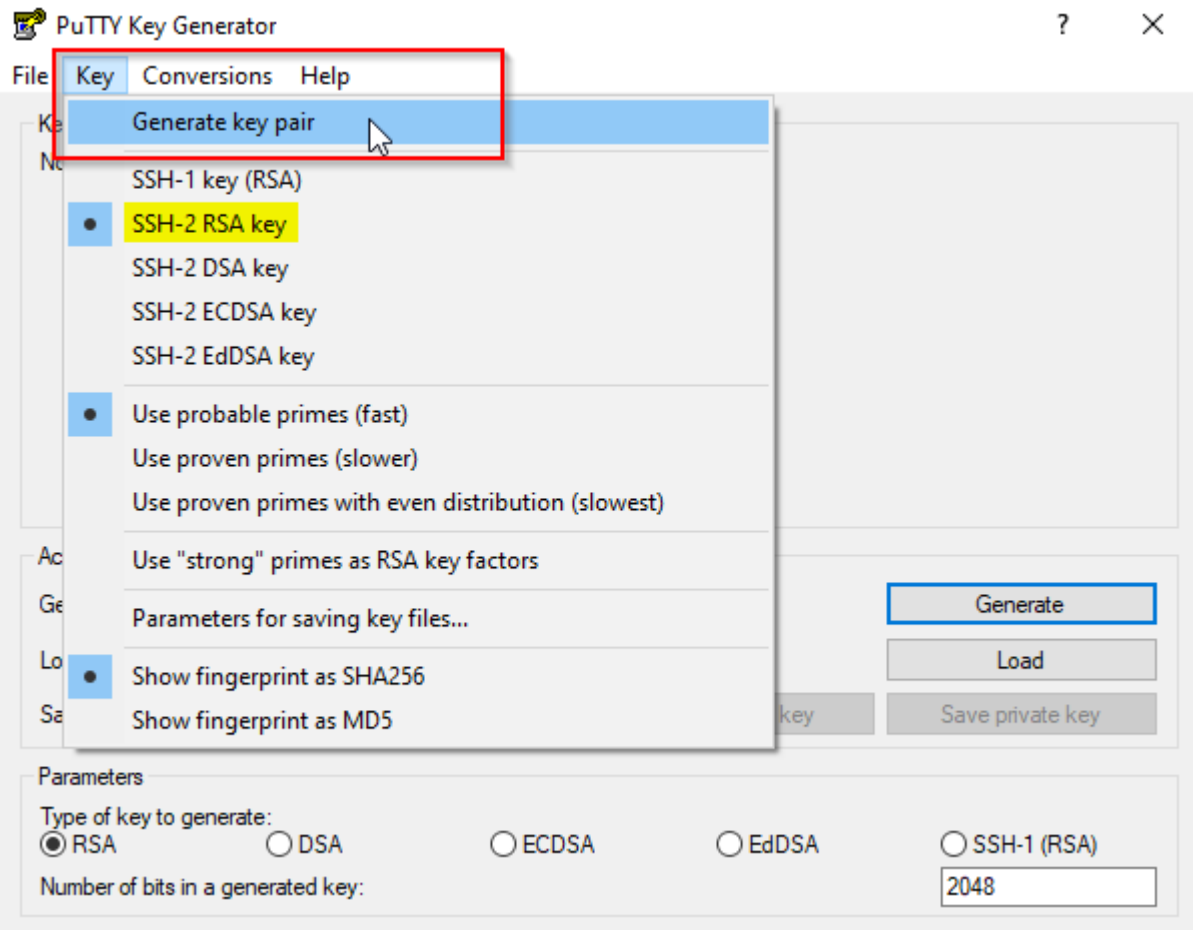
# USE PUTTYGEN TO MAKE SSH KEYS

*Contributor: Randy Smith WU2S*

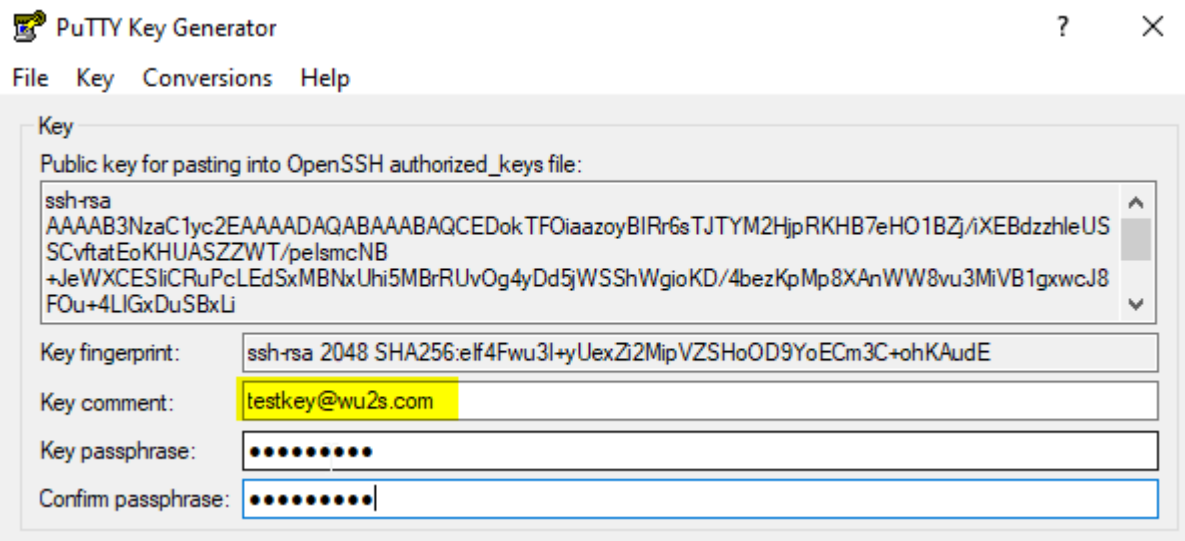
This How-to will show you a method for generating SSH key pairs on a Windows computer, saving them to a USB flash drive, installing the SSH key on an AREDN® node and using the SSH keys with a PuTTY terminal session. The use of Secure Shell (SSH) keys when using PuTTY or another SSH client is a useful aid to managing a group of AREDN® nodes.

- First, obtain the PuTTY suite of applications from the PuTTY Download Page and install them on your computer.
- Second, obtain and prepare to use a text editor such as Notepad++ that allows you to remove unwanted characters and metadata from your key file.
- Finally, follow the steps below to create, edit, and install your SSH keys.

1. Start the PuTTYGen application. Confirm that you are going to generate an SSH-2 RSA key.

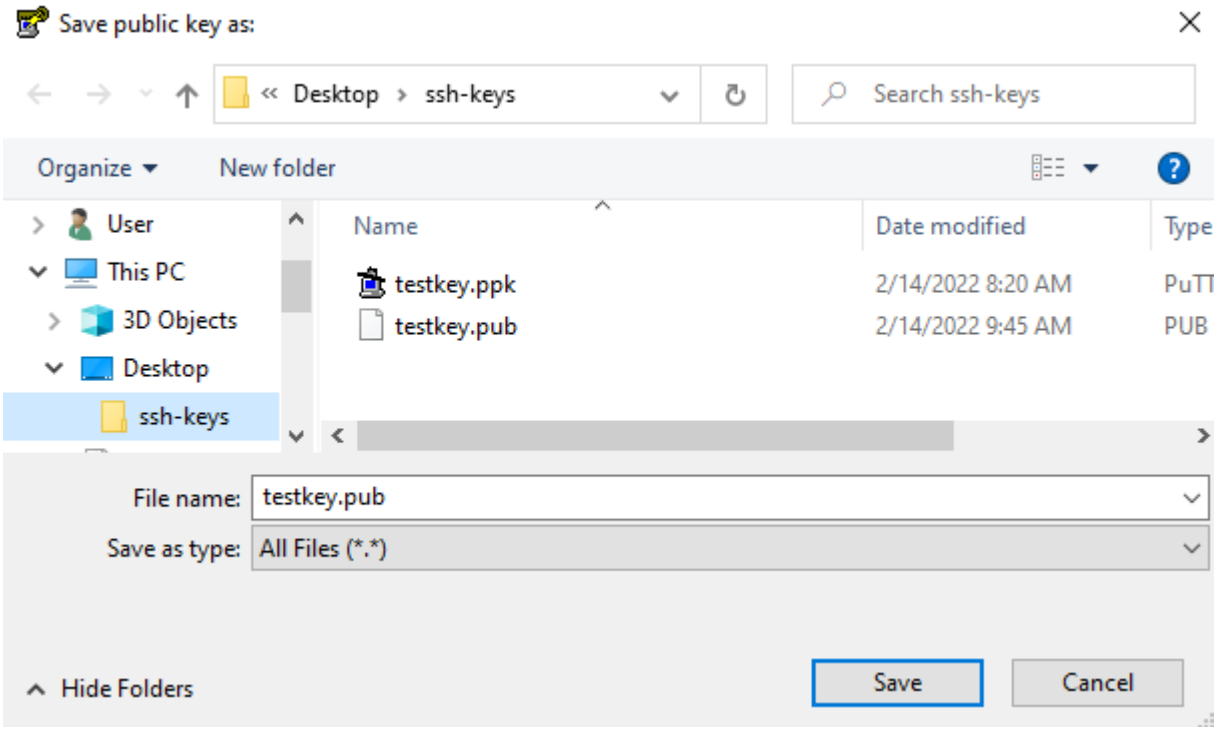



2. Select the *Generate key pair* menu item or click the *Generate* button and you will be asked to make some random mouse movements. After a short while you get a message asking you to wait while the keys are generated. Once it finishes you now have a new key pair.

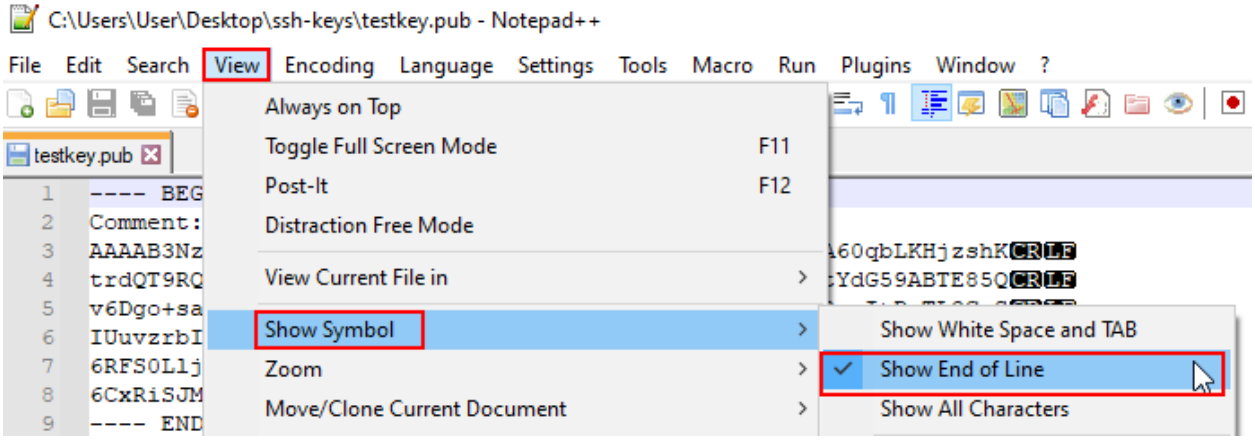



Give the key pair a suitable comment so that you will remember what the keys are used for. Here we just entered testkey@wu2s.com for an example. Whatever you enter in the "Key Comment" field must look like an email address with no spaces and the "@" present. Normally this field is used to identify a specific *username@hostname*. You can also password protect the SSH login by providing a passphrase if you desire. Record this passphrase so you will remember it for future use.

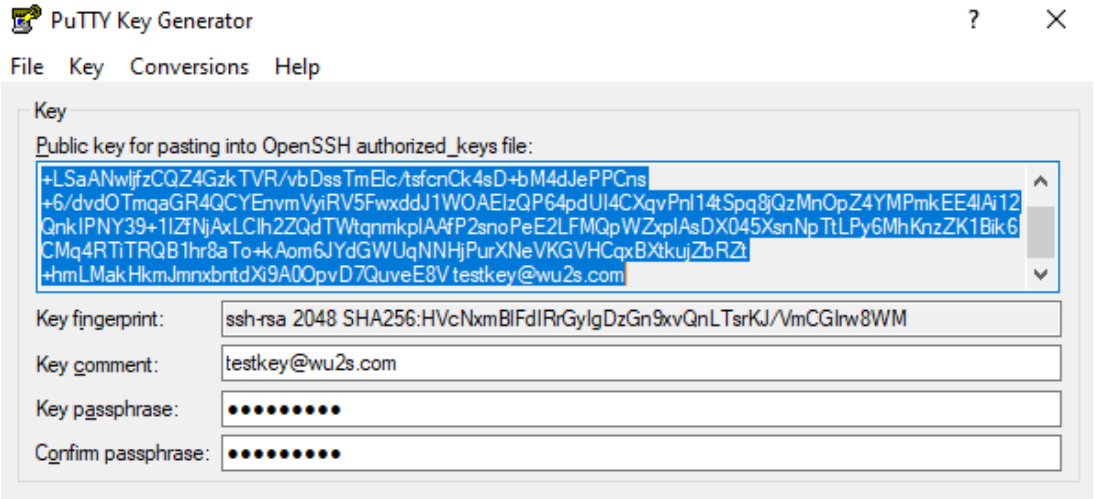
3. In PuTTYGen you can save your new keys to separate files for later use. To save the public key to a suitable location, click the *Save Public Key* button and enter a filename with a **.pub** extension. Then click the *Save Private Key* button to save your private key to the same location. Give your private key a **.ppk** file extension. Many people save their keys on a USB flash drive to maintain physical possession of them at all times.



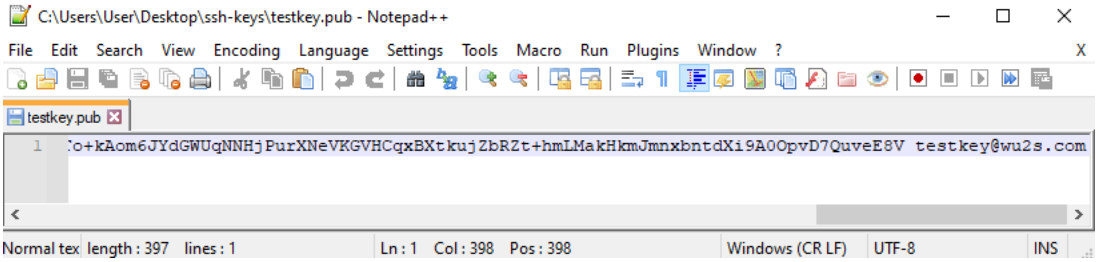

4. In order for your new public key to be installed on an AREDN® node you will need to verify that there are no extra characters which Windows typically adds to text files. You can accomplish this using a text editor which allows you to view and remove the unwanted characters. This example shows opening Notepad++ and navigating to *View* > *Show Symbol* > *Show End of Line*. Now you can see the line termination characters inserted by Windows.



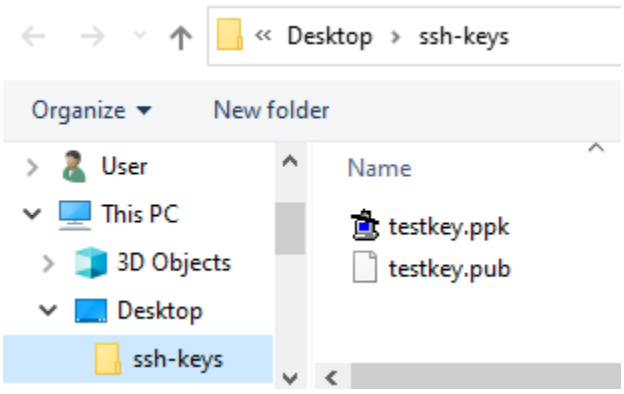
If you saved your public key file by clicking the *Save Public Key* button in PuTTYGen you may notice that it contains a header, footer, and lots of end of line characters. Your AREDN® node will not accept the file with these extra characters. The easiest way to resolve this is to go back to PuTTYGen and highlight/select the entire contents of the text area titled "Public key for pasting into OpenSSH authorized_keys file." Copy this text using the CTRL-C keys on your keyboard.



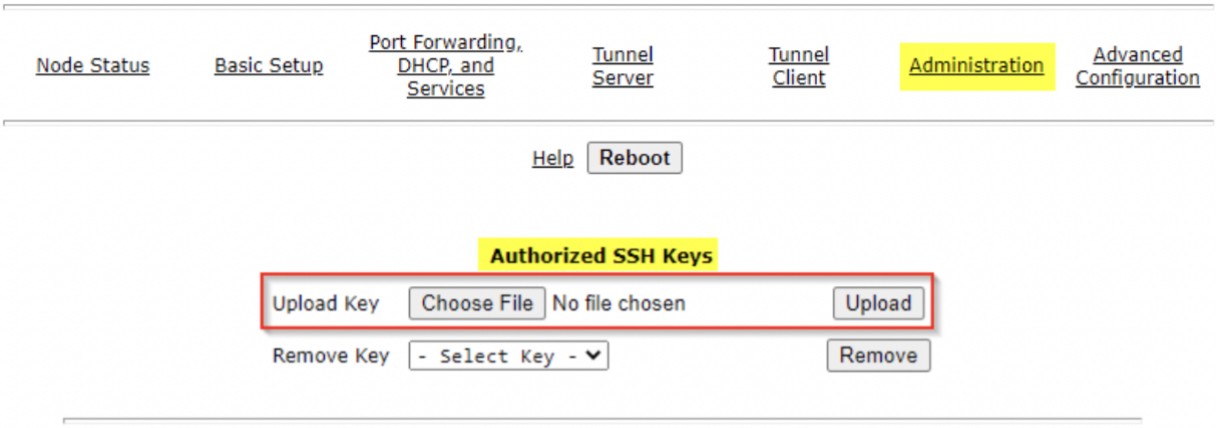

Now go to Notepad++ and paste the copied text into a new window. You should see your public key text on a single line without any header/footer or line termination characters.



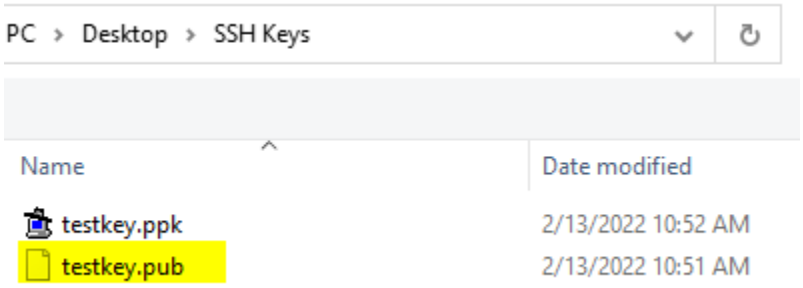

Save this Notepad++ window to a suitable filename with the **.pub** file extension.




5. In order to use your new SSH key pair, login to your AREDN® node and go to the **Setup -> Administration** screen. At the bottom you will see the *Authorized SSH Keys* section where you can install the public keys to use on this node.

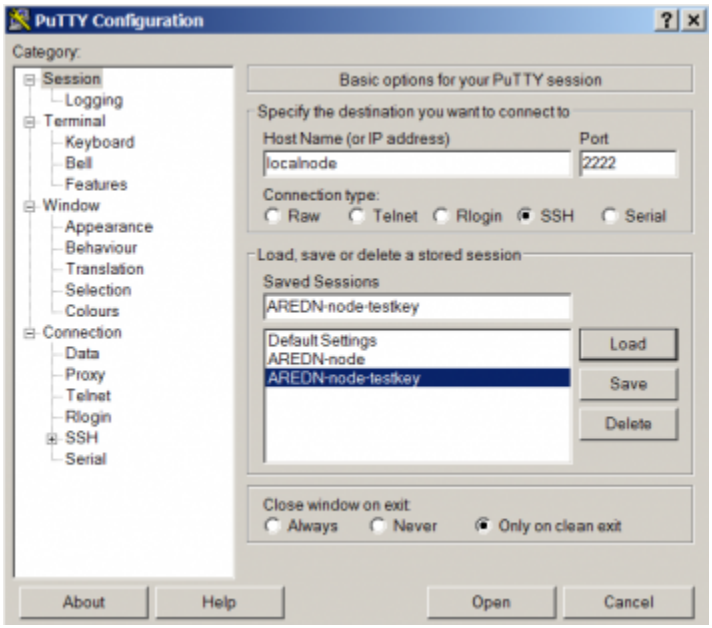



6. Press the *Choose File* button to locate the *public* SSH key you want to install. After choosing the desired *public* key file, click the *Upload* button to install the key on the AREDN® node.

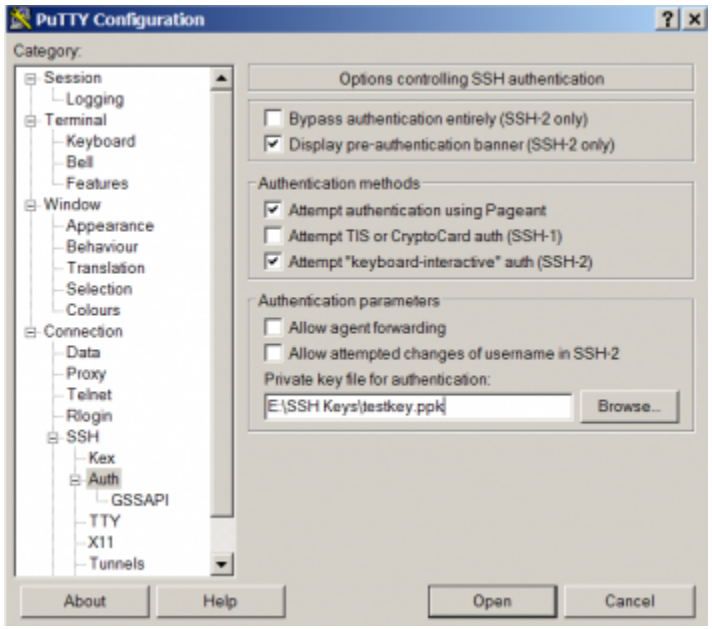



7. You will see a message asking you to reboot your node. After rebooting you can confirm that the new key was installed by looking in the dropdown list under the *Remove Key* section. Your SSH key will appear in the list if it is installed. (You are verifying that the key was installed, but do not click the *Remove* button unless you want to remove it.)


8. To use your SSH keys, open a new PuTTY session. In the *Hostname* box enter *localnode* and in the *Port* box enter 2222. It may be helpful to save this session definition using a name that identifies the specific node you are connecting to. Enter your identifier and click the *Save* button.

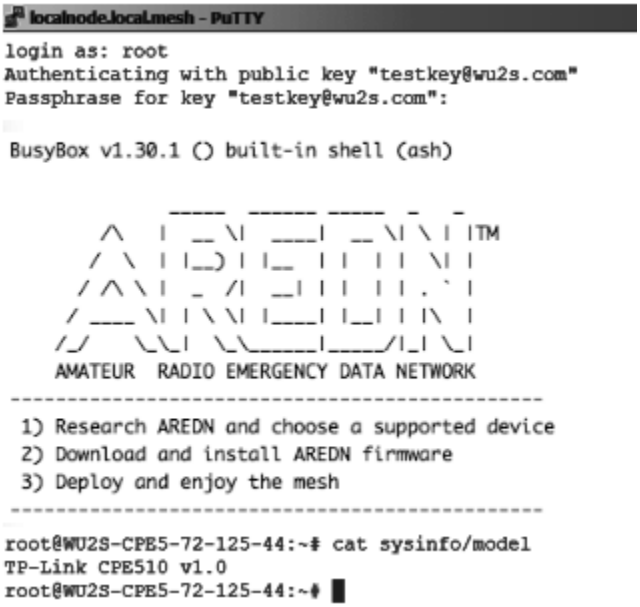



9. Now, using the menu at the left, go to the SSH section and then select the *Auth* item. This shows a number of Options. The only one we need is the very last – the location of the Private key file for authentication. Browse for it and select the correct filename as before. Remember that the PRIVATE key files end in .ppk Go back to top of the menu on the left and select *Session*. SAVE the session definition again.



10. Now you can use the session information you saved by clicking the *Load* or *Open* button in the main PuTTY session screen. This will open a terminal window as shown below. Login to the AREDN® node as *root*. If you configured the PuTTY session correctly, it will find your private key file and ask you for the passphrase (if any). If PuTTY cannot find the private key file, it will revert to prompting you for the *root* password that you normally use to login on the node.


11. The correct passphrase was entered. The node's banner appears in the terminal session window and you can now do any command line tasks on the node.



Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

THIRTY

# CREATING A LOCAL PACKAGE SERVER

There may be cases where your mesh nodes have no way to access the AREDN® servers for installing new packages. One way to resolve this is to create your own package 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 package 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 package repository.

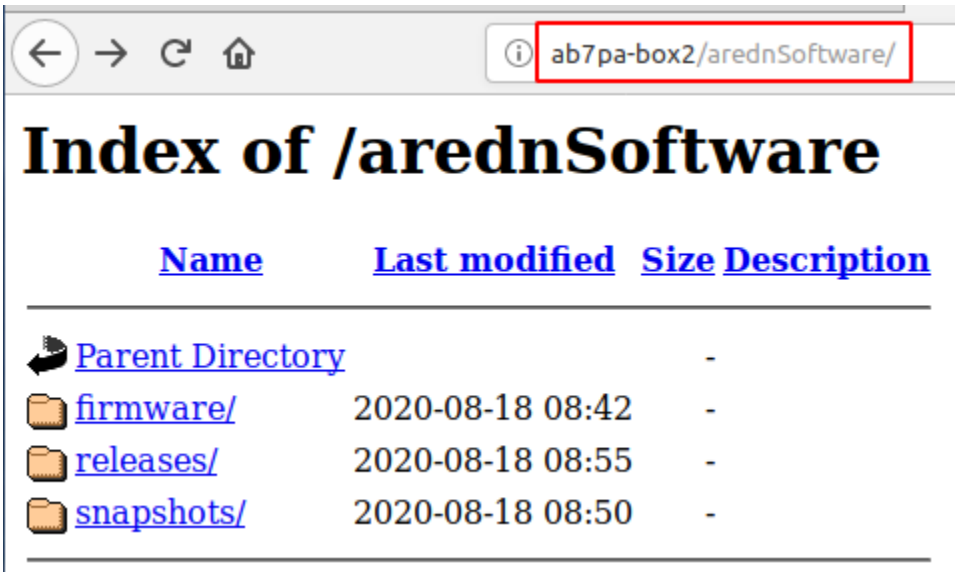
## 30.1 Configure your Package Server

Your package 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® Internet server, but the node's Internet connection is not advertised or allowed for use by other nodes or devices on the mesh network. You should add this host to the node's *DHCP Reservation List*. You do not need to add the package host to the *Advertised Services List* of the node to which it is connected. The package 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 package server:

- Obtain the set of AREDN® software files from `downloads.arednmesh.org`
- Make those files available via your computer's web server so nodes can query the package URLs

There are several ways to accomplish these tasks, and the best approach may vary depending on the platform you implement for your package server. Downloading the AREDN® software files can be done manually as needed, or the process could be automated and executed on a regular schedule. Tools that could be used for this task include HTTrack and Wget, both of which support recursive copying. You should try to make your local repository mirror the AREDN® downloads directory tree as closely as possible, so it contains any of the package files you want to have available to your local mesh nodes.

Once you have downloaded the AREDN® files, you need to make them 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, Sys Admins using the Apache Web Server might put the software files under their web server's *DocumentRoot*, or they might 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 to navigate the entire package tree as shown below.




These tasks are all that should be required on your local package host. Once the package tree is available via its web server, you can begin pointing the nodes to your local software repository.

## 30.2 Point Nodes to the New Server

To point a node to the local software repository, navigate to **Setup > Advanced Configuration**. The table on this webpage has a row for each type of software that can be installed on AREDN® nodes. It might be a good idea to take a screenshot of these settings so you can refer to them later. A typical default URL for *firmwarepath* is shown below:

```
http://downloads.arednmesh.org/firmware
```

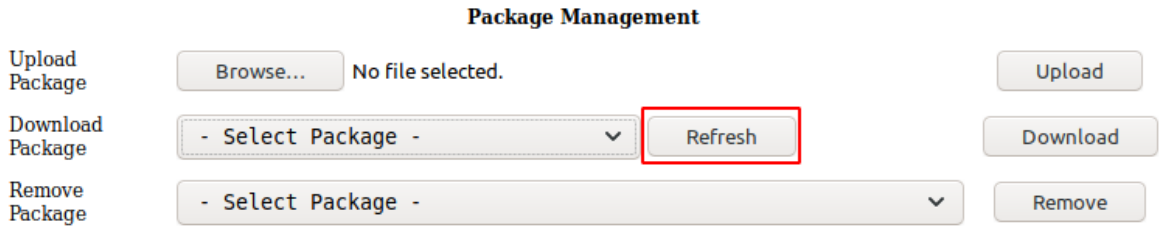
Simply replace this URL with the one that you configured on your local package host, then click the *Save Setting* button on that row. For example, the new entry for *firmwarepath* might look like the one below:

```
http://ab7pa-box2.local.mesh/arednSoftware/firmware
```

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 exactly with the alias or path you created and tested on your web server as described in the previous section.


After you have entered the new URL, click the **Save Setting** button to activate the new entry. To restore the default entry, click the **Set to Default** button.

Once the node has been pointed to the local package repository, you can navigate to **Setup > Administration**. In the *Package Management* section, you can click the **Refresh** button to get the list of available packages from the local package repository. Remember that retrieving this package list will use memory resources on your node.




The following example shows the type of information returned when you click the **Refresh** button:

```
Package Management

Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
 ↪mips_24kc/base/Packages.gz
Updated list of available packages in /var/opkg-lists/aredn_base
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
 ↪mips_24kc/base/Packages.sig
Signature check passed.
```

(continues on next page)

(continued from previous page)

```
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
 ↪mips_24kc/arednpackages/Packages.gz
Updated list of available packages in /var/opkg-lists/aredn_arednpackages
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
 ↪mips_24kc/arednpackages/Packages.sig
Signature check passed.
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
 ↪mips_24kc/luci/Packages.gz
Updated list of available packages in /var/opkg-lists/aredn_luci
Downloading http://ab7pa-box2.local.mesh/arednSoftware/snapshots/packages/
 ↪mips_24kc/luci/Packages.sig
Signature check passed.
...
```

Click the **Select Package** dropdown list to see the packages that are available for download to your node. Select a package and click the **Download** button. Status information will appear showing the actions that were taken to install the package from the local package host. A message may appear that a reboot is required to refresh and restart all services, but this is a normal status message and does not indicate an error condition.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# THIRTYONE

# TOOLS FOR DEVELOPERS

This section of the AREDN® documentation contains information useful for developers who want to retrieve information from one or more nodes for use in any of several applications. For example, a developer may want to write a program which periodically polls 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. The popular KG6WXC MeshMap application uses these tools to create and update a comprehensive mesh network map.

## 31.1 SYSINFO.JSON

The **sysinfo.json** API (Application Programming Interface) has been included in AREDN® firmware for several releases, and each release 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/cgi-bin/sysinfo.json`

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 RF* 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": "1.8",
"lat": "33.101010",
"lon": "-101.101010",
"grid_square": "DM22xx",
"node": "CALLSIGN-NODE-22",
"sysinfo": {
  "uptime": "5 days, 6:22:30",
  "loads": [
    0.05003,
    0.05003,
    0
    ]
  },
"node_details": {
  "description": "CALLSIGN-22 node information here...",
  "mesh_gateway": "0",
  "model": "MikroTik RouterBOARD 952Ui-5ac2nD ",
  "board_id": "0x0000",
  "firmware_mfg": "AREDN",
  "firmware_version": "1101-ad0caaf"
  }
}
```


In addition to the basic information described above, which is always returned with every invocation, the **sysinfo.json** 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.json?hosts=1&services=1` will include both the *hosts* and *services* information in addition to the basic details which are always returned.

### 31.1.1 Add Hosts Information

To retrieve mesh hosts information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?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"
  }
],
...
```

### 31.1.2 Add Services Information

To retrieve mesh services information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?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",
    "protocol": "tcp",
```

(continues on next page)

(continued from previous page)

```
    "link": "http:\/\/MYCALL-NODE-81\/iperfspeed"
  },
  {
    "name": "EtherPad",
    "protocol": "tcp",
    "link": "http:\/\/MYCALL-RPI:9001\/"
  },
  {
    "name": "MeshChat",
    "protocol": "tcp",
    "link": "http:\/\/MYCALL-RPI\/meshchat"
  }
],
...
```

### 31.1.3 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/cgi-bin/sysinfo.json?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.

### 31.1.4 Add Link Information

To retrieve mesh link information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?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 TUN (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": {
  "10.22.22.22": {
    "hostname": "CALLSIGN-NODE-22",
```

(continues on next page)

(continued from previous page)


```
    "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"
  }
},
...
```


Link: AREDN Webpage

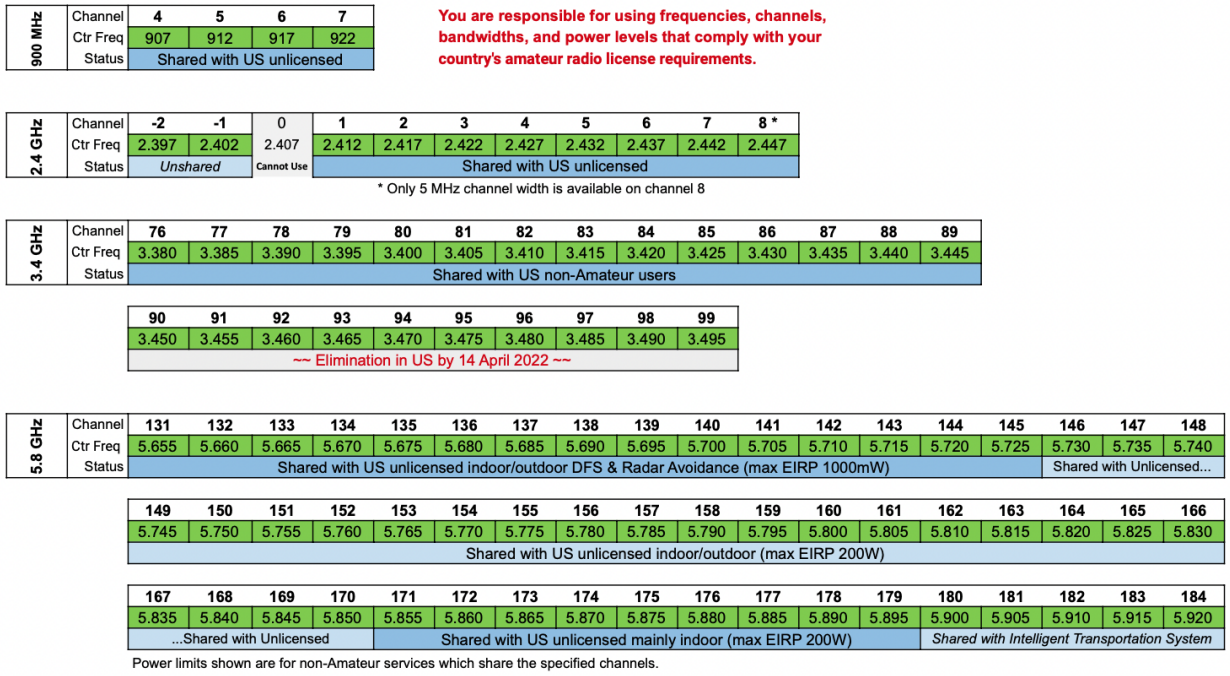
Link: AREDN Webpage

CHAPTER

# THIRTYTWO

# FREQUENCIES AND CHANNELS

Example US frequencies and channels that are available for AREDN® networking are shown in the diagram below.



**900 MHz**

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

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

**2.4 GHz**

| Channel | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 * |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ctr Freq | 2.397 | 2.402 | 2.407 | 2.412 | 2.417 | 2.422 | 2.427 | 2.432 | 2.437 | 2.442 | 2.447 |
| Status | *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 | Shared with US non-Amateur users | | | | | | | | | | | | | |

| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
|---|---|---|---|---|---|---|---|---|---|
| 3.450 | 3.455 | 3.460 | 3.465 | 3.470 | 3.475 | 3.480 | 3.485 | 3.490 | 3.495 |
| ~~ Elimination in US by 14 April 2022 ~~ | | | | | | | | | |

**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 (max EIRP 1000mW) | | | | | | | | | | | | | | | 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 (max EIRP 200W) | | | | | | | | | | | | | | | | | |

| 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 (max EIRP 200W) | | | | | | | | | *Shared with Intelligent Transportation System* | | | | |

Power limits shown are for non-Amateur services which share the specified channels.

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# THIRTYTHREE

# ADDITIONAL INFORMATION

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

- AREDN homepage
- AREDN forums

## 33.1 Contributing AREDN® 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. 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.*

Link: AREDN Webpage

Link: AREDN Webpage

CHAPTER

# THIRTYFOUR

# LICENSE

## 34.1 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.

### 34.1.1 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.

### 34.1.2 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.

### 34.1.3 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.

### 34.1.4 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.

### 34.1.5 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.

### 34.1.6 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.

### 34.1.7 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.

### 34.1.8 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.

### 34.1.9 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.

### 34.1.10 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.*

Link: AREDN Webpage

Link: AREDN Webpage