
AREDN Documentation

Release 3.19.3.0

AREDN

Nov 06, 2019

Getting Started Guide

1	AREDN® Overview	3
2	Selecting Radio Hardware	5
3	Downloading AREDN® Firmware	7
4	Installing AREDN® Firmware	9
5	Basic Radio Setup	17
6	Node Status Display	21
7	Mesh Status Display	27
8	Advanced Configuration	31
9	Networking Overview	45
10	Network Topologies	47
11	Radio Spectrum Characteristics	51
12	Channel Planning	57
13	Network Modeling	65
14	AREDN Services Overview	69
15	Chat Programs	73
16	Email Programs	81

17 File Sharing Programs	85
18 VoIP Audio/Video Conferencing	89
19 Video Streaming and Surveillance	97
20 Computer Aided Dispatch	105
21 Other Possible Services	109
22 Firmware Upgrade Tips	115
23 How-to Use PuTTYGen on Windows to Make SSH Keys and Use Them on AREDN® Nodes	117
24 Settings for Radio Mobile	127
25 Test Network Links with iperf	131
26 Frequencies and Channels	133
27 Additional Information	135



Version 3.19.3.0

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.

AREDN® is a registered trademark of *Amateur Radio Emergency Data Network, Inc.*

CHAPTER 1

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.

CHAPTER 2

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 features 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 or version of radio, and the current list of supported devices is found on the AREDN® website in the [Supported Platform Matrix](#).

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 the 900 MHz, 2.4 GHz, 3.4 GHz, and 5.8 GHz bands.
- Many devices come with an integrated dual-polarity MIMO (Multiple Input-Multiple Output) antenna which helps to mitigate multipath propagation issues.
- 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 \$50 to several hundred dollars for a complete node, 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 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 device information is a manufacturer's datasheet, usually available for download from the manufacturer's website. Currently AREDN® supports over fifty device models from the following manufacturers: 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 any of several kinds of high-gain antennas.

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

CHAPTER 3

Downloading AREDN® Firmware

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.

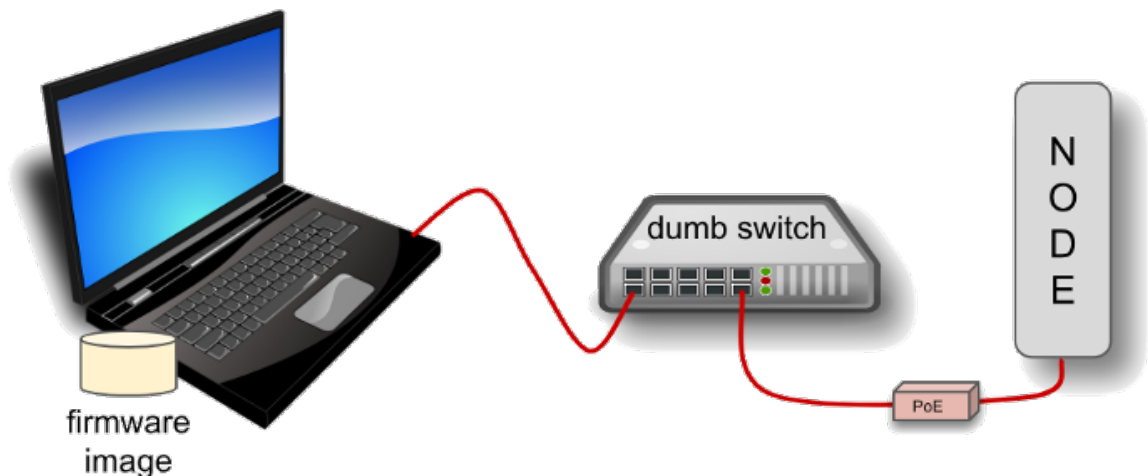
CHAPTER 4

Installing AREDN® Firmware

The steps for installing device firmware are documented on the AREDN® website in the [Current Software](#) section. Under the **Software** menu, select **Download** to reach the *Current Software* page.

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 interface to navigate to **Setup > Administration > Firmware Update** to install your new firmware. This process will be explained in more detail in the **Advanced Configuration** section of this guide.
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 link while the node is being rebooted.

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 TP-LINK devices, your computer's web browser will connect to the node's web server to upload the firmware image. For Mikrotik devices, your computer will run a remote boot server and the node's remote boot client will load its boot image from your computer. Refer to the specific procedures below for your node hardware.

4.1 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 a [TFTP command line client](#).

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 of 192.168.1.5 with a netmask of 255.255.255.0
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.
4. 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.
5. Open a command window on your computer and execute the following commands to send the AREDN firmware to your device:

```
>>>
> tftp 192.168.1.20    [If your device is an AirRouter use 192.
↪168.1.1]
> bin                 [This puts the transfer in the required
↪"binary" mode]
> trace on            [This will show the transfer in_
↪progress]
```

(continues on next page)

(continued from previous page)

```
> put <full path to the AREDN firmware file>
    [For example, put c:\temp\aredn-3.19.3.0-ubnt-nano-m-
    ↪xw-factory.bin]
```

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

6. 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.
7. Configure your computer's Ethernet network interface to use DHCP for obtaining an IP address from the node.
8. After the node reboots, open a web browser and enter the following URL: `http://localnode.local.mesh:8080`
9. Navigate to the *Setup* page and configure the new “firstboot” node as described in the **Basic Radio Setup** section.

4.2 TP-LINK First Install Process

4.2.1 Preferred Process

TP-LINK devices currently allow you to use the manufacturer's pre-installed *PharOS* web browser user interface to upload and apply new firmware images. This is the most user-friendly way to install AREDN® firmware. Navigate to the *Setup* section 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.

4.2.2 Alternate Process

TP-LINK devices also have a built-in TFTP (Trivial File Transfer Protocol) and **Bootp** client which allows them to obtain new firmware from an external source. Your computer must run a TFTP/Bootp server in order to provide firmware images to the node. In certain situations you may need to use this method to update the firmware or to restore a TP-LINK recovery file by following the steps below.

Preparation

1. Download the appropriate TP-LINK *factory* file and rename this file as `recovery.bin`
2. Set your computer's Ethernet network adapter to a static IP address of 192.168.0.100 with a netmask of 255.255.255.0

3. 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.

Linux Procedure

1. Create a directory on your computer called `/tftp` and copy the TP-LINK `recovery.bin` file there.
2. Determine your computer's Ethernet interface name with `ifconfig`. It will be the interface you set to 192.168.0.100 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below.
3. Become `root` and open a terminal window to execute the following `dnsmasq` command:

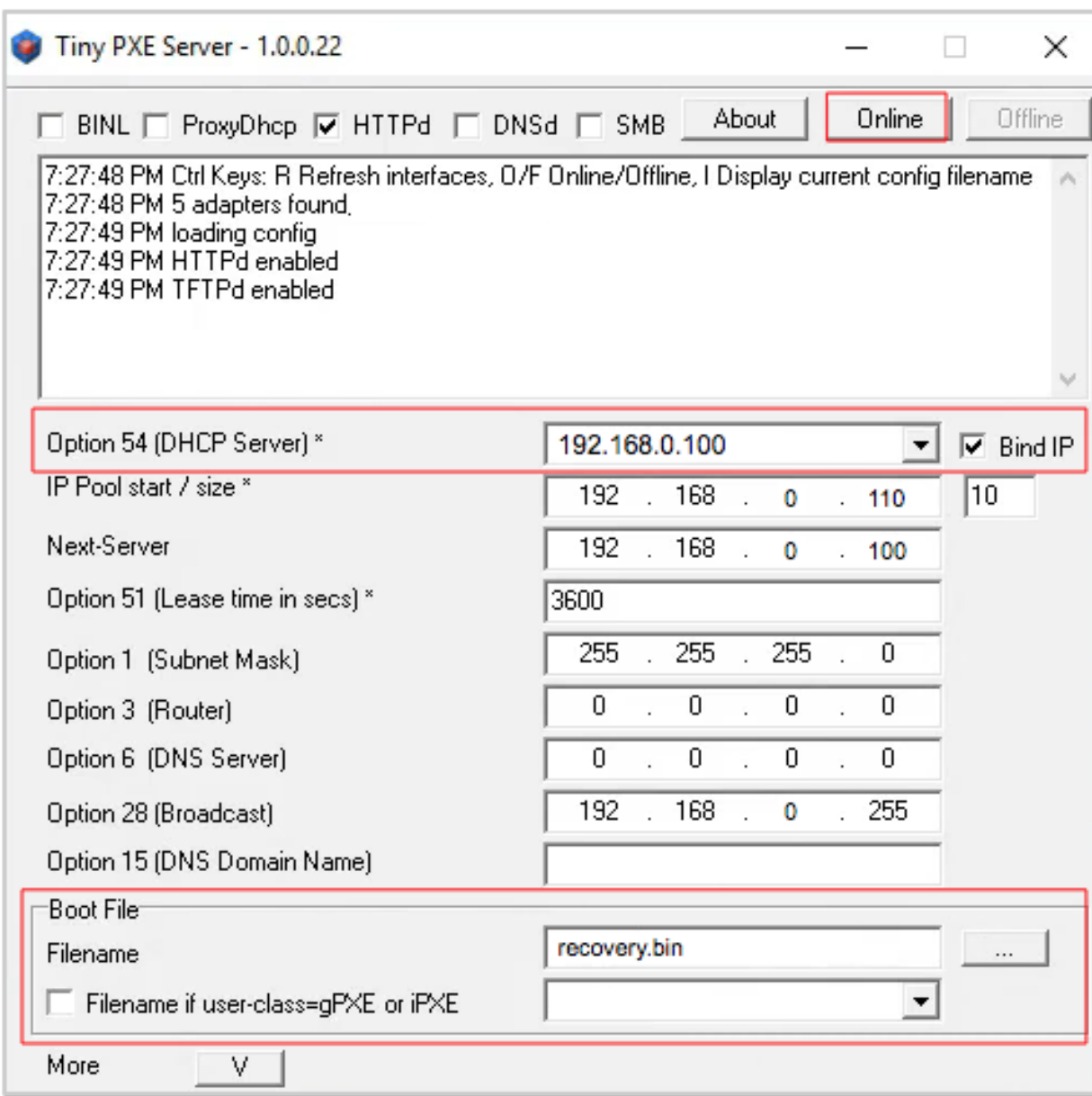
```
>>>
# dnsmasq -i eth0 -u joe --dhcp-range=192.168.0.150,192.168.0.200 \
--dhcp-boot=recovery.bin --enable-tftp --tftp-root=/tftp/ \
-d -p0 -K --log-dhcp --bootp-dynamic
```

4. With the PoE unit powered off, connect an Ethernet cable from the TP-LINK node to the POE port.
5. Push the reset button on the TP-LINK and hold it while powering on the PoE unit. 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 about 10 seconds. Release the reset button as the computer starts communicating with the node. When you see the “sent” message, this indicates success, and the TP-LINK node has downloaded the image and will reboot. You can now `<ctrl>C` or kill `dnsmasq`.

Windows Procedure

You will need [Tiny PXE](#) software on your Windows computer. Download this software and extract it on your computer.

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 Ethernet interface IP 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 `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 PoE unit powered off, connect an Ethernet cable from the TP-LINK node to the POE port. Press and hold the reset button on the node while powering on the PoE unit.
5. Continue holding the reset button until you see TFTPd: DoReadFile: recovery.bin in the Tiny PXE log window.
6. 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.

Final Configuration Steps

1. Configure your computer's Ethernet network interface to use DHCP for obtaining an IP address from the node.

2. After the node reboots, open a web browser and enter the following URL: `http://localnode.local.mesh:8080`
3. Navigate to the *Setup* page and configure the new “firstboot” node as described in the **Basic Radio Setup** section.

4.3 Mikrotik First Install Process

Mikrotik devices must be flashed using steps that are similar to the alternate TP-LINK process described above. Your computer must run a TFTP/Bootp server in order to provide firmware images to Mikrotik nodes. Mikrotik nodes require a **two-part install** process: First, install and boot the *factory* (elf) file, and finally use the in-memory-only AREDN® Administration UI to complete the installation of the *sysupgrade* (bin) file.

Preparation

1. Download the appropriate Mikrotik *factory* and *sysupgrade* files. Rename the factory file to `rb.elf` and keep the *sysupgrade* file available for later.
2. Set your computer's Ethernet network adapter to a static IP address of 192.168.1.10 with a netmask of 255.255.255.0
3. 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. If you are flashing a Mikrotik hAP ac lite device, connect the Ethernet cable from Port 1 of the Mikrotik to the dumb switch.

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 192.168.1.10 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below.
3. Become `root` and open a terminal window to execute the following `dnsmasq` command:

```
>>>
# dnsmasq -i eth0 -u joe --dhcp-range=192.168.0.100,192.168.0.200 \
--dhcp-boot=rb.elf --enable-tftp --tftp-root=/tftp/ \
-d -p0 -K --log-dhcp --bootp-dynamic
```

4. With the PoE unit powered off, connect the Mikrotik node to the POE port. Press and hold the reset button on the Mikrotik while powering on the PoE unit or the hAP device.
5. 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 about 10 seconds. Release the reset button as the computer starts communicating with the node. When you see the

“sent” message, this indicates success, and the node has downloaded the image and will reboot. You can now <ctrl>C or kill dnsmasq.

Windows Procedure

You will need [Tiny PXE](#) software on your Windows computer. Download this software and extract it on your computer.

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 Ethernet interface IP 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 `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 PoE unit powered off, connect the Mikrotik node to the POE port. If you are flashing a Mikrotik hAP ac lite device, connect the LAN cable from Port 1 of the Mikrotik to the dumb switch.
5. Press and hold the reset button on the node while powering on the PoE unit or the device. Continue holding the reset button until you see `TFTPd: DoReadFile: rb.elf` in the Tiny PXE log window.
6. 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.

Final Configuration Steps

1. After booting the AREDN firmware image the node should have a default IP address of 192.168.1.1. Change your computer’s Ethernet interface to DHCP mode to obtain an IP address from the node. You should be able to ping the node at 192.168.1.1. If this does not work, then something is wrong. Don’t proceed until you can ping the node.
2. In a web browser, open the node’s Administration page <http://192.168.1.1:8080/cgi-bin/admin> (user = ‘root’ password = ‘hsmm’) and navigate to the *Setup > Administration > Firmware Update* section. Select the *sysupgrade* file you previously downloaded and click the *Upload* button.
3. After the node reboots, navigate to the node’s *Setup* page and configure the new “firstboot” node as described in the **Basic Radio Setup** section.

Once your device is running AREDN® firmware, you can display its web interface by connecting your computer to the LAN port on the PoE and navigating to the following URL: `http://localnode:8080`

By default AREDN® devices run the DHCP (Dynamic Host Control Protocol) service on their LAN interface, so your computer will receive an IP address from the node as soon as it is connected with an Ethernet cable. Ensure that your computer is set to obtain its IP address via DHCP.

CHAPTER 5

Basic Radio Setup

After you have installed the AREDN® firmware, rebooted the device, and connected your computer to the LAN port on the POE you can navigate to the following URL: `http://localnode:8080`. The initial status page will be displayed, instructing you to configure your node by clicking the **Setup** button.



NOCALL-22-15-88

Location Not Available

[Help](#)

[Refresh](#)

[Setup](#)

[Select a theme ▼](#)

This node is not yet configured.
 Go to the setup page and set your node name and password.
 Click Save Changes, even if you didn't make any changes, then the node will reboot.

WiFi address	192.168.2.1 / 24	firmware version	3.19.3.0
LAN address	none	configuration	not set
WAN address	none	system time	Fri Mar 01 2019 07:56:50 UTC
default gateway	none	uptime	5 min
SSID	N/A	load average	0.08, 0.41, 0.24
Channel	11	free space	flash = 1552 KB /tmp = 13912 KB memory = 5488 KB
Bandwidth	Mhz	OLSR Entries	Total = 0 Nodes = 0

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.

[Help](#)

Node Name

Password

Node Description (optional)

Verify Password

Mesh RF	LAN	WAN	
Enable <input checked="" type="checkbox"/>	LAN Mode <input type="button" value="5 host Direct"/>	Protocol <input type="button" value="DHCP"/>	
IP Address <input type="text" value="10.22.15.88"/>	IP Address <input type="text" value="10.176.122.193"/>	DNS 1 <input type="text" value="8.8.8.8"/>	
Netmask <input type="text" value="255.0.0.0"/>	Netmask <input type="text" value="255.255.255.248"/>	DNS 2 <input type="text" value="8.8.4.4"/>	
SSID <input type="text" value="AREDN"/>	DHCP Server <input checked="" type="checkbox"/>	<div style="text-align: center;">Advanced WAN Access</div>	
Channel <input type="button" value="-20-v3"/>	DHCP Start <input type="text" value="194"/>	Allow others to use my WAN <input type="checkbox"/>	
Channel Width <input type="button" value="20 MHz"/>	DHCP End <input type="text" value="198"/>	Prevent LAN devices from accessing WAN <input type="checkbox"/>	
<div style="text-align: center;">Active Settings</div>			
Tx Power <input type="button" value="26 dBm"/> ?			
<div style="display: flex; justify-content: space-between;"> 0.00 miles </div>			
Distance to FARTHEST Neighbor <input type="text" value="0"/>			
<div style="display: flex; justify-content: space-between;"> 0 kilometers </div>			
<div style="display: flex; justify-content: space-between;"> 0 meters </div>			
<input type="button" value="Apply"/>			

In order to get your new AREDN® node on the air, 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 beaconsed 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-1. This is similar to the MYCALL setting you would give a packet TNC (Terminal Node Controller), but without the 0-15 character restriction.

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

- 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.
- 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.

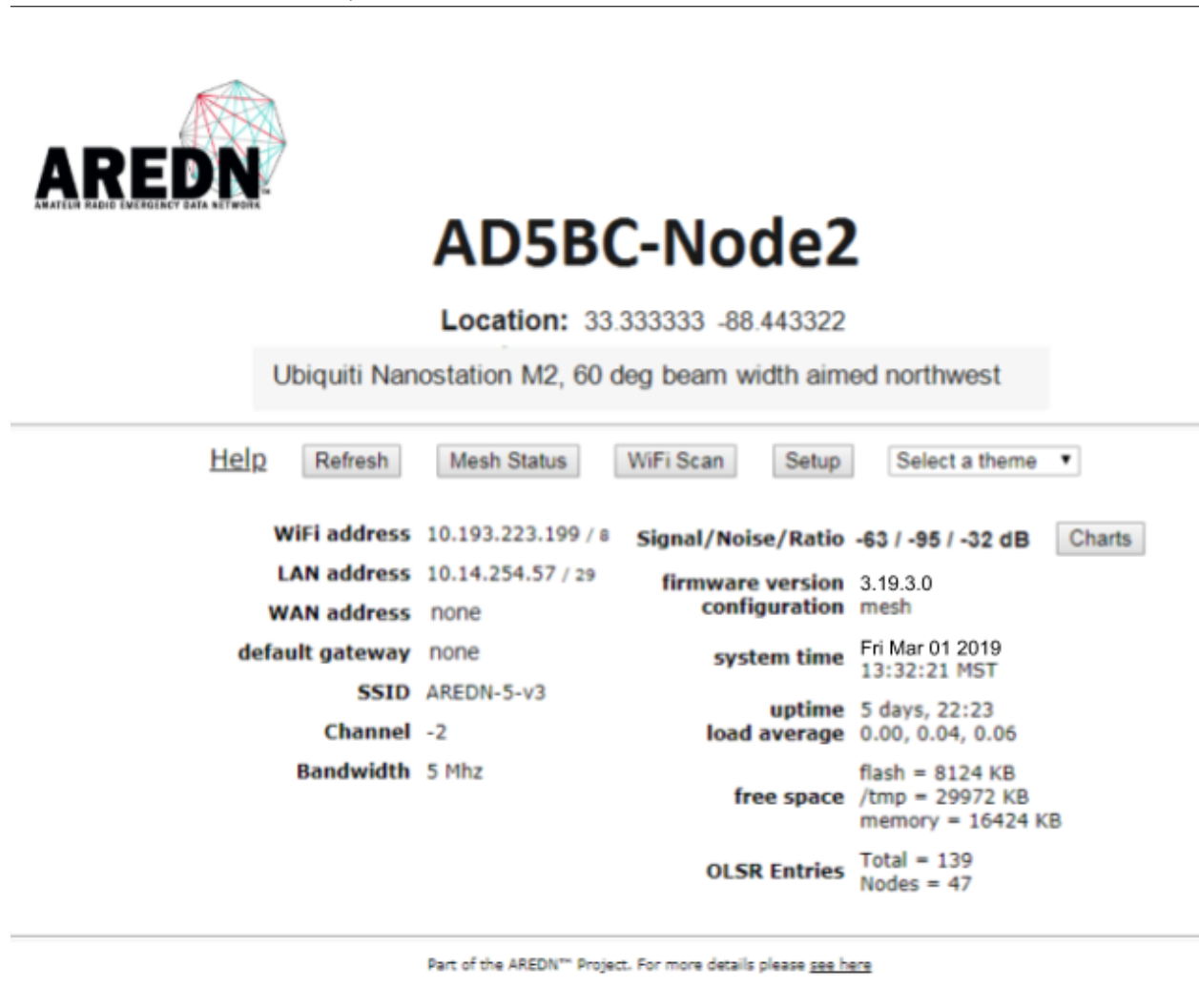
Optional Settings In this section you can enter your node's latitude and longitude, as well as the grid square designator. Click the **Apply Location Settings** button after entering this information. You may also change the timezone for your node's system time.

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.

CHAPTER 6

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:8080`. You will be redirected to the **Node Status** page as shown below.



AD5BC-Node2

Location: 33.333333 -88.443322

Ubiquiti Nanostation M2, 60 deg beam width aimed northwest

Help Refresh Mesh Status WiFi Scan Setup Select a theme ▼

WiFi address	10.193.223.199 / 8	Signal/Noise/Ratio	-63 / -95 / -32 dB	Charts
LAN address	10.14.254.57 / 29	firmware version	3.19.3.0	
WAN address	none	configuration	mesh	
default gateway	none	system time	Fri Mar 01 2019 13:32:21 MST	
SSID	AREDN-5-v3	uptime	5 days, 22:23	
Channel	-2	load average	0.00, 0.04, 0.06	
Bandwidth	5 Mhz	free space	flash = 8124 KB /tmp = 29972 KB memory = 16424 KB	
		OLSR Entries	Total = 139 Nodes = 47	

Part of the AREDN™ Project. For more details please [see here](#)

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), but only if they are using the same bandwidth settings as your node. When multiple ad-hoc networks are visible (with different SSIDs or channels), the *network* is displayed but not the individual nodes. There is also an automatic scan mode, but running a wifi scan continuously is not recommended because this will degrade mesh performance. A wifi scan transmits queries on all channels to discover other devices.

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](#).

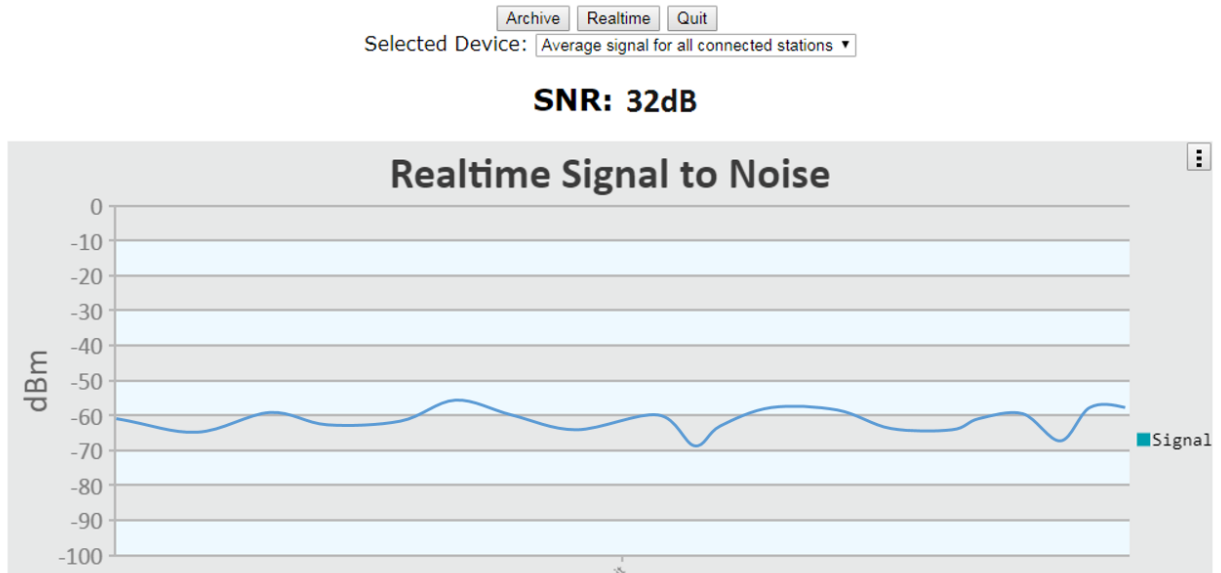
Below the Signal Strength readings are the node's **Firmware Version** and network type. The **System Time** is displayed, as well as the **Uptime**, or time since the last reboot. Nodes have no internal battery or realtime clock, so the time is reset every time the node is booted. If an Internet connection 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 **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.

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.

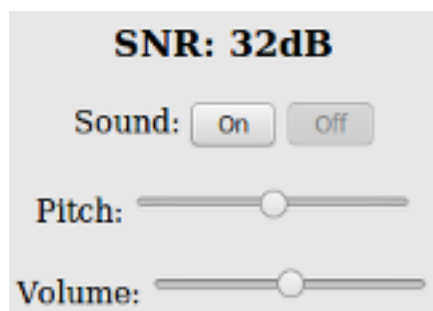
Below these controls you can choose to view the signal strength statistics for individual nodes that are directly connected to your node. Choose the neighbor node from the **Selected Device** dropdown list. Changing the selected device will automatically reload the chart to show that node's information.

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.

Mesh Status Display

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

AD5BC-Node2 mesh status

Location: 33.333333 -88.443322

Ubiquiti Nanostation M2, 60 deg beam width aimed northwest

RefreshAutoQuit

Local Hosts	Services	Current Neighbors	LQ	NLQ	TxMbps	Services
AD5BC-Node2.local.mesh		AD5XX-Tunnel-Server.local.mesh	100%	100%		meshchat
		● AD5XX-services-host				

Remote Nodes	ETX	Services	Previous Neighbors	When
AD5YY-2.local.mesh	1.10		none	
AD5ZZ-TACNODE.local.mesh	1.10			

OLSR Total = 12
Entries Nodes = 4

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.

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 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.
- **TxBps** 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

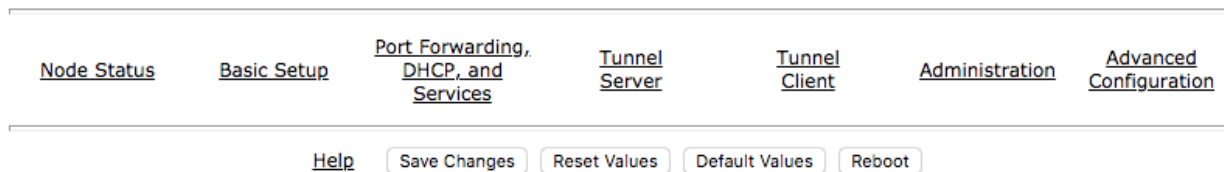
the OLSR protocol selects the least cost route based on the lowest ETX path in the direction of the final destination.

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.

CHAPTER 8

Advanced Configuration

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 several 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.

Node Name	AD5BC-Node2		Password	
Node Description (optional)	Ubiquiti Nanostation M2 with integrated 60 deg dual polarity antenna aimed northwest		Verify Password	

Mesh RF	LAN	WAN
Enable <input checked="" type="checkbox"/>	LAN Mode 5 host Direct ▼	Protocol Static ▼
IP Address 10.22.15.88	IP Address 10.176.122.193	IP Address 192.168.10.10
Netmask 255.0.0.0	Netmask 255.255.255.248	Netmask 255.255.255.0
SSID AREDN -5- v3	DHCP Server <input checked="" type="checkbox"/>	Gateway 192.168.10.1
Channel -2 (2397) ▼	DHCP Start 194	DNS 1 8.8.8.8
Channel Width 5 MHz ▼	DHCP End 198	DNS 2 8.8.4.4
<hr/> <div> Active Settings </div> <div> Tx Power 26 dBm ▼ ⓘ </div> <div> 3.11 miles </div> <div> Distance to FARTHEST Neighbor 5 kilometers </div> <div> 5000 meters </div> <div> <input type="range"/> </div> <div> <input type="button" value="Apply"/> </div>		
<div> Advanced WAN Access </div> <div> Allow others to use my WAN <input type="checkbox"/> </div> <div> Prevent LAN devices from accessing WAN <input type="checkbox"/> </div>		

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.

The *Distance* 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 return. If the distance parameter is too long, the transmitter will wait extra time before considering the data lost and retransmitting the packets. The *Distance* setting is only applicable to nodes that can communicate directly over RF.

Mesh RF		LAN	
Enable	<input type="checkbox"/>	LAN Mode	5 host Direct ▼
IP Address	10.22.15.88	IP Address	10.176.122.193
Netmask	255.0.0.0	Netmask	255.255.255.248
		DHCP Server	<input checked="" type="checkbox"/>
		DHCP Start	194
		DHCP End	198
<hr/>			
LAN Access Point			
Enable	<input checked="" type="checkbox"/>	SSID	AD5BC-AREDN
Channel	7 ▼	Encryption	WPA2 PSK ▼
Password	••••••••••		

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](#).

If you enabled the *LAN Access Point* feature, 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 assign the node a *Static* IP address on your WAN network. 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.

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.

DHCP Address Reservations				Advertised Services			
Hostname	IP Address	MAC Address		Name	Link	URL	
ad5bc-host2	10.14.254.61	54:ab:3a:04:58:a4	Del	meshchat	<input checked="" type="checkbox"/>	http://ad5bc-host2:8080/meshchat	Del
	- IP Address -		Add		<input type="checkbox"/>	://AD5BC-Node2:	Add

Current DHCP Leases			
ad5bc-host2	10.14.254.61	54:ab:3a:04:58:a4	Add

Port Forwarding				
Interface	Type	Outside Port	LAN IP	LAN Port
WAN	TCP		- IP Address -	

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 to the right of the host 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.

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 you 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.

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 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, the request will be routed to the LAN IP address and port on which that host is listening for incoming service requests.

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](#).

8.3 Tunnel Server

Click the **Tunnel Server** link to navigate to these settings. This section provides a way for you to configure your node with a special service that allows 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 install the *Tunnel Server* feature. This is because it will cause your node to dedicate limited system resources to running a service that may be used rarely. 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, so the network will 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. There are also AREDN® [website posts](#) which contain helpful information.

Your node should have Internet access after the smart switch is configured, and you can use the node's new Internet connection to install the *tunneling* software package. This package should be installed on both the tunnel server and the tunnel client nodes.

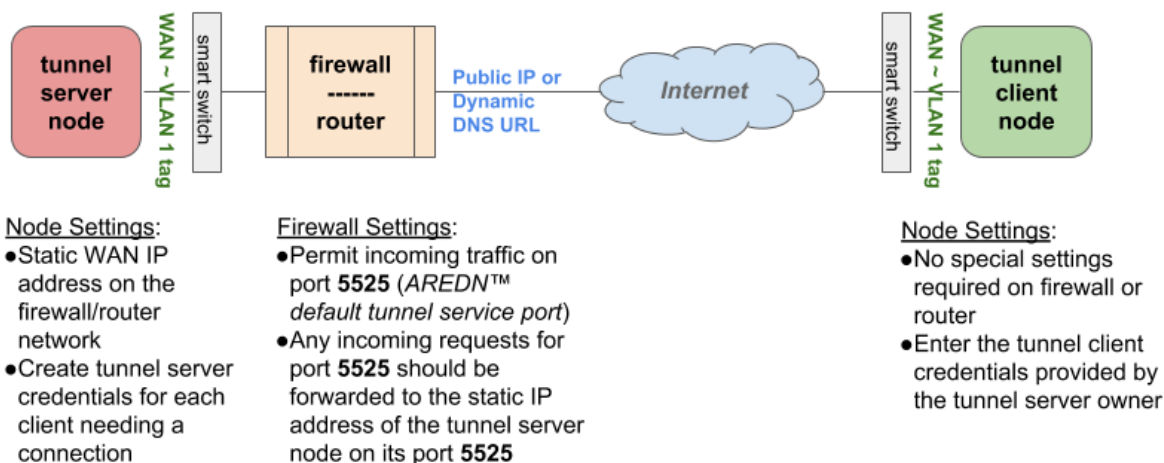
WAN Interface IP (Tunnel Server *node only*) Set a static IP address on your tunnel server node's WAN interface so your Internet-connected router/firewall has a consistent way to forward traffic to your node.

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.

8.3.2 Tunnel Server Node Settings

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

AREDN™ Tunnel Service Configuration



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.

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.

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.

Connect this node to the following servers:

Enabled?	Server	Pwd	Network	Active	Action
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="button" value="Add"/>

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.

8.5 Administration

Click the **Administration** link to navigate to these settings. There are four sections that provide a way for you to update the AREDN® firmware, as well as to install or remove software packages on your node.

Firmware Update

current version: 3.19.3.0

Upload Firmware
No file selected.

Download Firmware

☒ Keep Settings

Package Management

Upload Package
No file selected.

Download Package

Remove Package

Authorized SSH Keys

Upload Key
No file selected.

Remove Key

Support Data

[Download Support Data](#)

Attention: Files cannot be uploaded to a node while a tunnel server or client connection is enabled. Disable tunnel client or server connections before uploading firmware, packages, or ssh key files. The *Upload* buttons will be disabled until tunnels are disabled.

Firmware Update If you have a new firmware image that has already been downloaded to your computer, click the *Browse* button and select the firmware file to upload. Click *Upload* and the file will be uploaded and installed on the node.

If the node has Internet access (either from its WAN interface or from the mesh) you can use the *Download Firmware* option. Click *Refresh* to update the list of available images. Select the image to download, click *Download*, and wait for the firmware to download and be installed. When upgrading firmware, you can retain your existing configuration settings by selecting the *Keep Settings* checkbox.

Package Management Here you can install or remove software packages on the node. *Upload Package* allows you to install a package file from your computer. *Download Package* allows you do retrieve a package over the Internet from the AREDN® website. Clicking *Refresh* will update the list of packages available for download, but try to avoid updating this list

unless you absolutely require it. The package information database is stored locally and will use quite a bit of storage space. Under normal circumstances it is rare to require a package refresh.

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.

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® forum 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 shown below. These settings are best left as default unless you have a clear understanding of why the defaults will not work for your node or mesh network.

Help Reboot Reset to Firstboot			
Help (hover)	Config Setting	Value	Actions
?	aredn.@map[0].maptiles	<input type="text" value="http://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token=pk.eyJ1Ijpc"/>	Save Setting Set to Default
?	aredn.@map[0].leafletcss	<input type="text" value="http://cdn.leafletjs.com/leaflet/v0.7.7/leaflet.css"/>	Save Setting Set to Default
?	aredn.@map[0].leafletjs	<input type="text" value="http://cdn.leafletjs.com/leaflet/v0.7.7/leaflet.js"/>	Save Setting Set to Default
?	aredn.@downloads[0].firmwarepath	<input type="text" value="http://downloads.arednmesh.org/firmware/ubnt"/>	Save Setting Set to Default
?	aredn.@poe[0].passthrough	<input checked="" type="checkbox"/> ON	Save Setting Set to Default
?	aredn.@usb[0].passthrough	<input checked="" type="checkbox"/> ON	Save Setting Set to Default

Above the settings table there are links that allow you to 1) view the node help file, 2) reboot the node, or 3) reset the node to a firstboot or “NOCALL” configuration.

Specific values can be set for the following items. You may change these settings and then click the *Save Setting* button. You may also reset these items to their default values by clicking the *Set to Default* button.

Map Tiles Specifies the URL where map tiles can be found.

Leaflet CSS Specifies the URL where the Leaflet CSS file can be found.

Leaflet JS Specifies the URL where the Leaflet Javascript file can be found.

Firmware Download Path Specifies the URL from which AREDN® firmware files can be downloaded.

PoE Passthrough Specifies whether Power over Ethernet should be enabled on nodes with ports that support PoE passthrough.

USB Passthrough Specifies whether the USB port should be enabled on nodes having a USB port.

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 server**
- **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.

CHAPTER 9

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* 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 area or 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, the topology becomes more complicated and data traffic may be routed through multiple “hops” in order to reach its intended destination. This could result in increased latency on the network, with some network segments becoming almost unusable if application response time thresholds cannot met.

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.

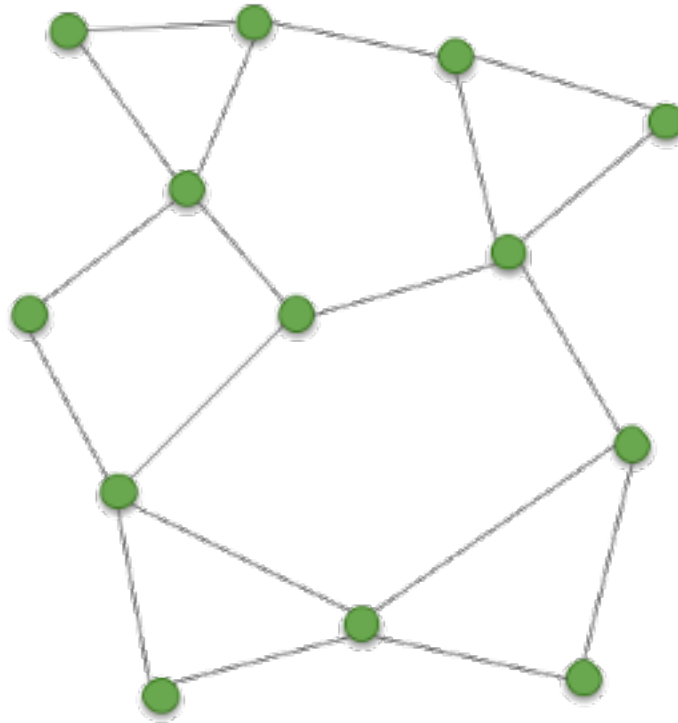
How many simultaneous users will be generating network traffic at different times? As the number of users increases, the amount of data traversing the network will also increase. In addition, with an increasing number of nodes on the network there will be a corresponding increase in the amount of OLSR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

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

CHAPTER 10

Network Topologies

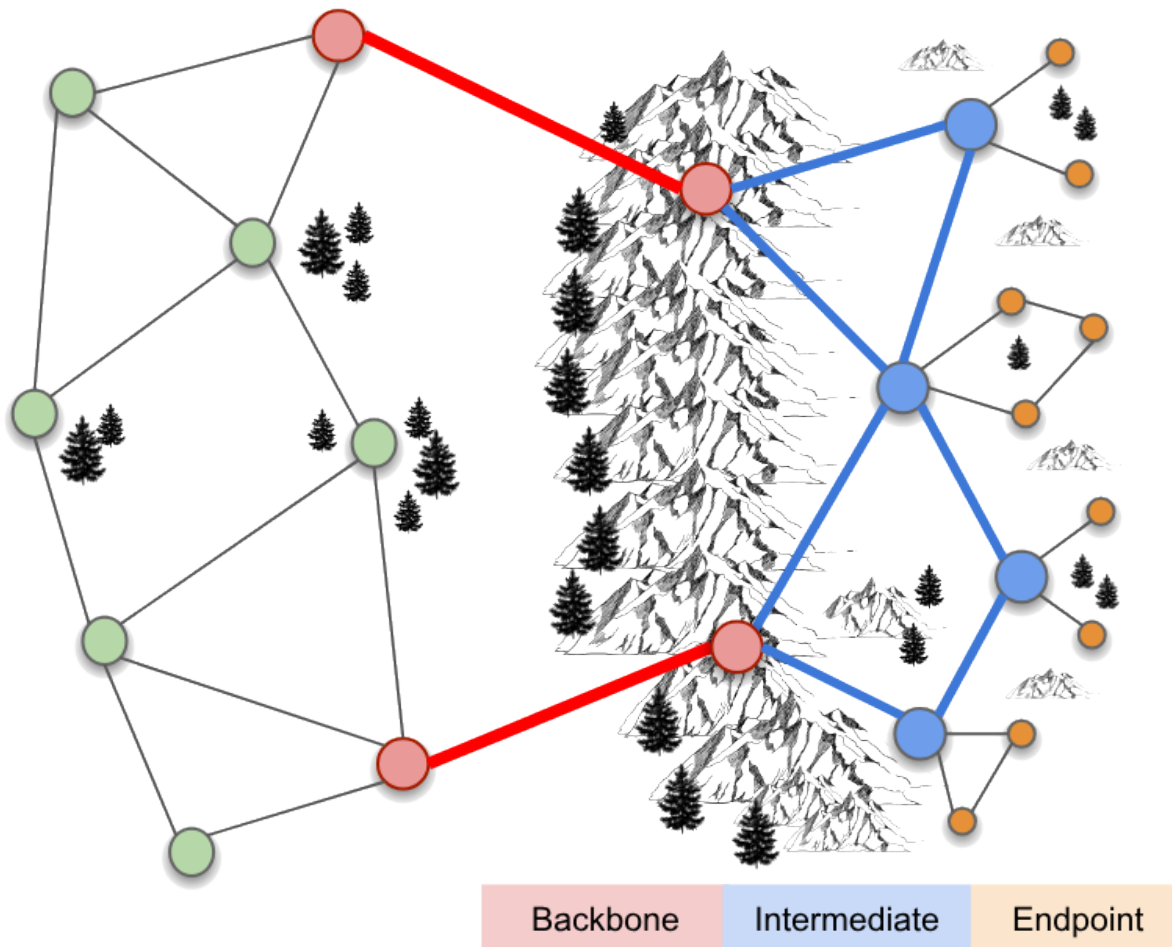
Every AREDN® node is capable of automatically joining an AREDN® mesh network which is operating with the same SSID, channel, and bandwidth. A *Mesh* topology consists of independent nodes which each explore their surroundings by broadcasting their identity and listening for their neighbors' responses. 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 routes based on changes in signal quality or loss of a link, allowing the network to adapt to changing conditions. Since there are usually several possible routes between nodes, and since network disruptions typically effect only part of the network, a *Mesh* topology can be self-healing.



This automatic ability to form a mesh network is built into the AREDN® firmware on each node. Every 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 basic *meshnet* may serve its purpose perfectly for a short-term network deployment in support of a local event, or even for more permanent communication between nodes which are always within radio range.

10.1 Types of Links

A variety of factors could isolate groups of mesh nodes from each other. For example, distance, terrain, structures, or foliage may prevent some of the nodes from communicating via RF. For long-term or permanent deployments there may be a need for special types of network links that connect what are called mesh “islands.” A *link* consists of both sides of a radio path, including the two devices that communicate back and forth across that path.



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.

Intermediate Links Intermediate links bridge the gaps between endpoint nodes. Their primary purpose is to pass network data, but there may be cases where they also serve as mesh access nodes for users. Sometimes these links are called “mid-mile”, “distribution”, or “relay” 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 intermediate 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 mesh network. Sometimes these links are call “last mile”, “tactical”, or “terminal” nodes. Usually these nodes

serve either as the originator or the final destination 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 mesh nodes that are required in order to fulfill the purposes for your network. The ultimate goal is to have a reliable data network that accomplishes its purpose for providing services to the intended destinations and users.

CHAPTER 11

Radio Spectrum Characteristics

AREDN® networks operate in the microwave radio spectrum, and licensed amateur radio operators have unique access to many of these frequencies. For bands in which amateur operators share the spectrum, there is an increased chance for RF interference which may make certain frequencies unusable for AREDN® data networking. All of the 33 cm band is shared with other FCC authorized users. All of the upper channels on the 13 cm band are shared with standard FCC Part 15 WiFi (IEEE 802.11x) and FCC Part 18 ISM (Industrial, Scientific, Medical) users, as are all of the lower channels on the 5 cm band. The only frequency range which amateur operators do not currently share with non-licensed users is the 9 cm band, in which the US military may occasionally operate radio location units. The following table lists each amateur radio band, frequency range, total bandwidth of allocation, and the number of channels that are available for AREDN® networking.

Band	Frequency Range	Total Bandwidth	Channels
33 cm	902-928 MHz	25 MHz	5
13 cm	2390-2450 MHz	45 MHz	9
9 cm	3300-3500 MHz	120 MHz	24
5 cm	5650-5925 MHz	260 MHz	52

The table above shows that the 9 cm band has the most available bandwidth in its unshared channels, while the 5 cm band has the next largest amount of available bandwidth in unshared channels. 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.

Some of the advantages and disadvantages of each frequency range are explained in the sections below.

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 required antenna system and support structures may be larger and heavier than those of higher frequency systems, and 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 five radio channels can exist in this shared frequency range, depending on the channel width that is selected.

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
	Status	Ham Band			Shared Ham and ISM/WiFi Band					
	Freq	2.397	2.402	2.407	2.412	2.417	2.422	2.427	2.432	2.437

*Not available for use

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. Radio and antenna systems for this band are often smaller in size and less difficult to install. 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 (height Above Ground Level) 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 (IEEE 802.16) devices which are designed for the 3.65 GHz range.

3.4 GHz	Channel	76	77	78	79	80	81	82	83	84	85	86	87
	Status	Ham Band											
	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
		88	89	90	91	92	93	94	95	96	97	98	99
	Freq	3.440	3.445	3.450	3.455	3.460	3.465	3.470	3.475	3.480	3.485	3.490	3.495

Refer to your local band plan for coordination

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 little 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 and terrain in order to achieve clear line of sight between nodes.

Channel	133	134	135	136	137	138	139	140	141	142	143	144	145
Status	Ham Band shared with U-NII-2C/wifi/unlicensed												
Freq	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
Channel	146	147	148	149	150	151	152	153	154	155	156	157	158
Status	Ham Band shared with U-NII-3/wifi/unlicensed												
Freq	5.730	5.735	5.740	5.745	5.750	5.755	5.760	5.765	5.770	5.775	5.780	5.785	5.790
Channel	159	160	161	162	163	164	165	166	167	168	169	170	171
Status	Ham Band shared with U-NII-3/wifi/unlicensed												
Freq	5.795	5.800	5.805	5.810	5.815	5.820	5.825	5.830	5.835	5.840	5.845	5.850	5.855
Channel	172	173	174	175	176	177	178	179	180	181	182	183	184
Status	Ham Band												
Freq	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

Refer to your local band plan for coordination: ★ 5825 to 5850 Shared under Part 15.247 with a limited number of WISP operators and may be encountered at tower sites

Advantages One advantage for using the 5 cm band is that it contains 52 channels, and many of them at the upper end of the band are 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 52 available channels. Radio and antenna systems for this band are often smaller in size and less difficult to install. 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.

CHAPTER 12

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, and channel width. Beyond that requirement, however, you have quite a bit of freedom to select the radio channels that will ensure the highest signal quality for your network.

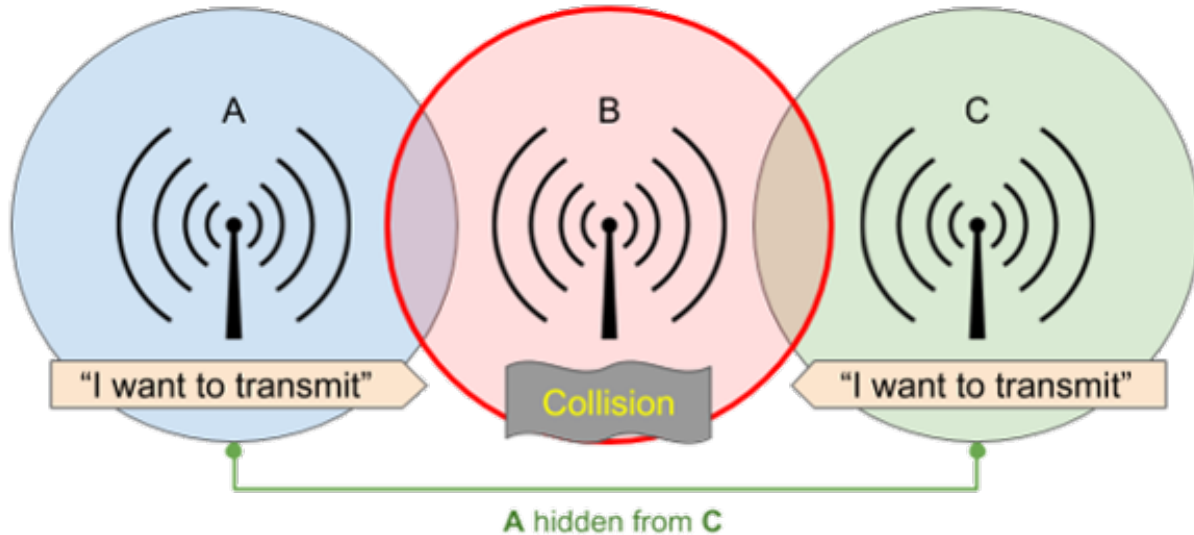
In a basic AREDN® *meshnet* with individual nodes spread across a limited geographical area, all of the nodes may use the same frequency, channel, and channel width. This allows them to establish a network and route data to any of the sites as needed. However, as more nodes join the network or when several nodes are COLLOCATED (sharing the same physical site), it is possible for them to interfere with each other. In order for an AREDN® network to scale up in size and complexity, channel planning becomes increasingly important.

Some amount of data traffic is required for OLSR to maintain and operate the mesh network. Having a growing number of nodes on the same channel will increase the amount of OLSR handshaking, which can lead to more latency for data traffic across the network. Any application, program, or service that is sensitive to latency (especially voice or video services) may experience difficulty or even become unusable.

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 links 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 establish a better link as an alternate path.

12.1 Channel Contention

In any wireless network there will be nodes which are not within radio range of each other. In the example below, **A** can hear **B** but cannot hear **C**. Since **A** and **C** are hidden from each other, they may try to transmit on the same channel at the same time without knowing it. Collision detection mechanisms will not help because the nodes have no way to communicate except through node **B**, so collision avoidance mechanisms must be used instead.



AREDN® firmware follows IEEE 802.11a/b/g/n standards and uses Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) for determining whether a channel is busy. Optionally AREDN® firmware also employs Request to Send / Clear to Send (RTS/CTS) messages to negotiate the use of a channel. Decreased network throughput is one side effect of this, because several nodes may be negotiating or waiting to use the channel.

For **hidden nodes**, two approaches may help to minimize this 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.

12.2 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.

12.3 Collocated Nodes



At some sites there may be several devices mounted on the same building or structure. The photo on the right shows many nodes collocated on a single tower. Network problems 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 rather than forcing them to use their radio channel.

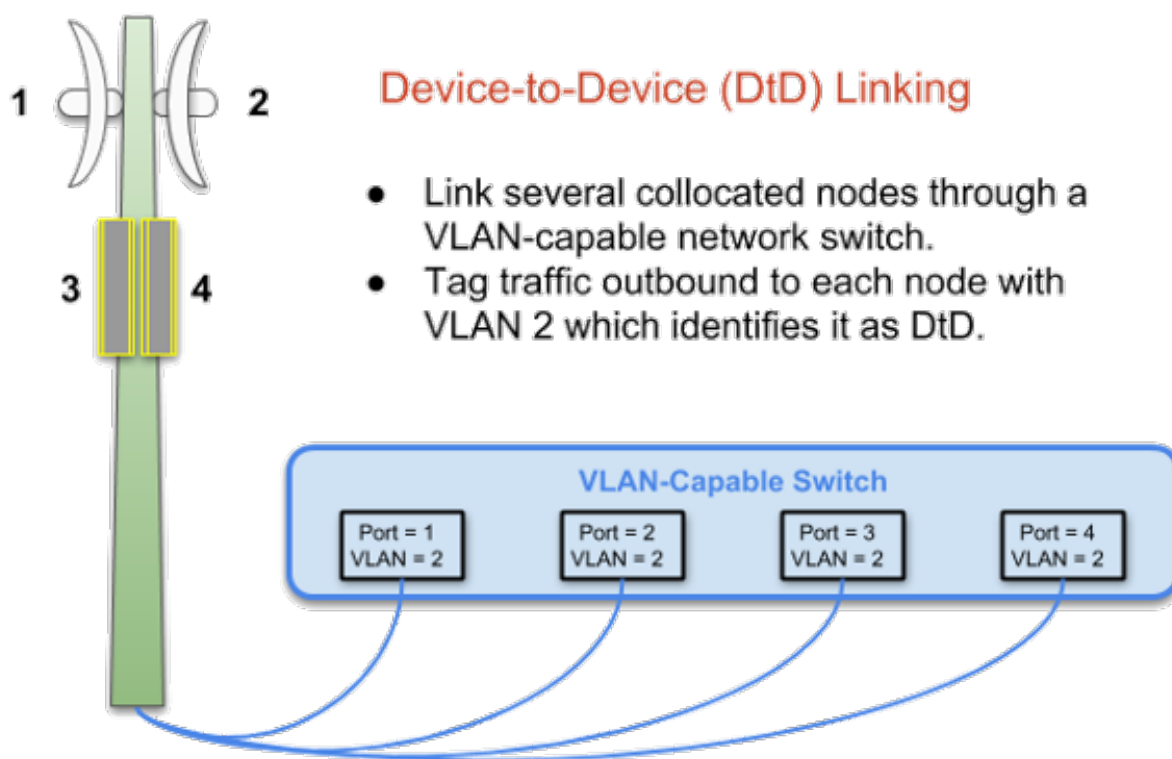
12.3.1 Device to Device (DtD) Linking

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

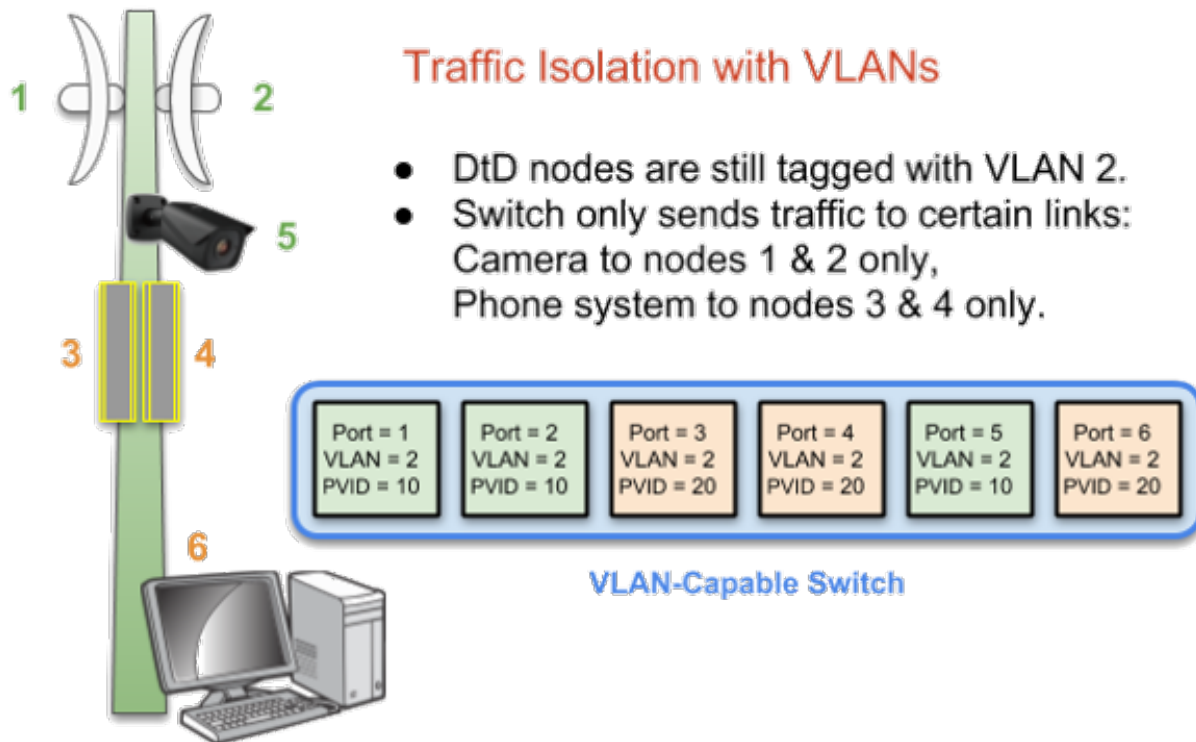
RF channel to become busy.

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 avoid interfering with each other 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, 3.4 GHz, and 5.8 GHz nodes to the same *meshnet*. 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 above, the smart switch will share all traffic across all ports and every node will receive it on its DtD link. If this is not what is desired, you can configure additional VLANs on the 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) phone system (6) and traffic from those devices should only be passed to a specific set of links as shown in the diagram below.

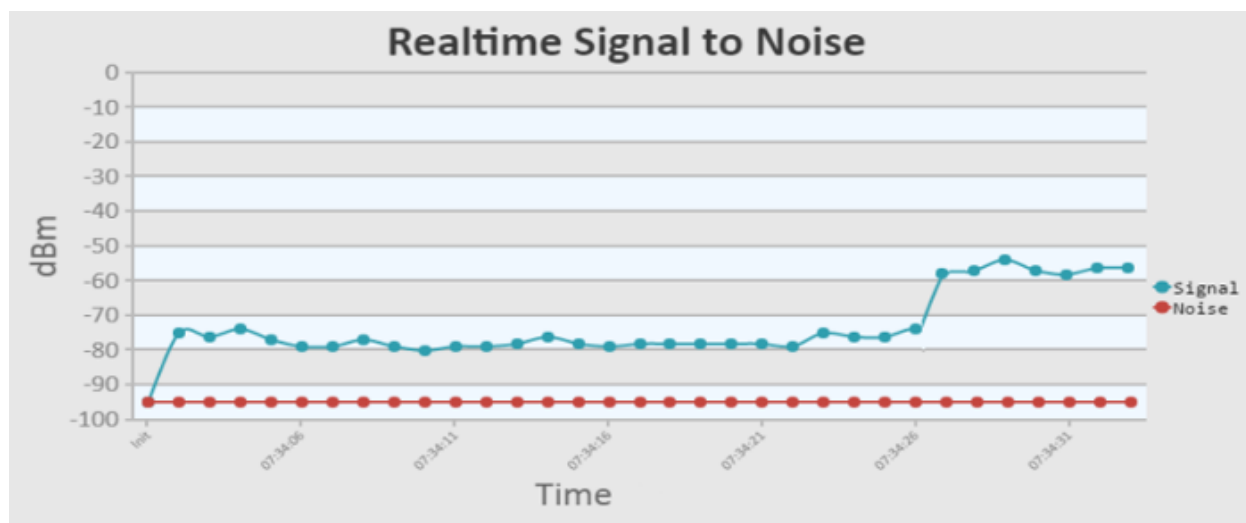


12.3.2 Antenna Polarization

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. Vertical polarization is usually preferred because it tends to be less susceptible to reflections and rain fade, but horizontal polarization still provides adequate signal with clear line of sight. Note that the antennas on both sides of a radio link must be oriented the same way.

12.4 Aligning Link 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 view 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).



12.5 Channel Planning Tips

Avoid Network Scalability Issues

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

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 link at half-duplex can send data at a time. This instantly cuts the throughput by one-third or more and increases latency with protocol overhead. To improve performance you can configure each link to use a different channel, allowing simultaneous transmissions. In the first case with channel sharing, it might be possible to have one HD video stream and one VoIP call with frequent dropouts. In the second case using different link channels, you could have three HD video streams and several VoIP calls simultaneously with few dropouts.

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 colocated nodes to avoid RF channel contention. The 3.4 GHz and 5.8 GHz bands provide the most unshared channels for use in AREDN® networks.

- 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 in order to avoid channel contention issues.
- Consider putting each local *meshnet* on its own channel to minimize the interaction between coverage areas, similar to how cellular network “cells” are planned and deployed.

- If you are installing long distance point to point links to connect mesh 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 focus on throughput without distractions.
- 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. The aggregate LQ should be at least 50% to have a usable path.

CHAPTER 13

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 two 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

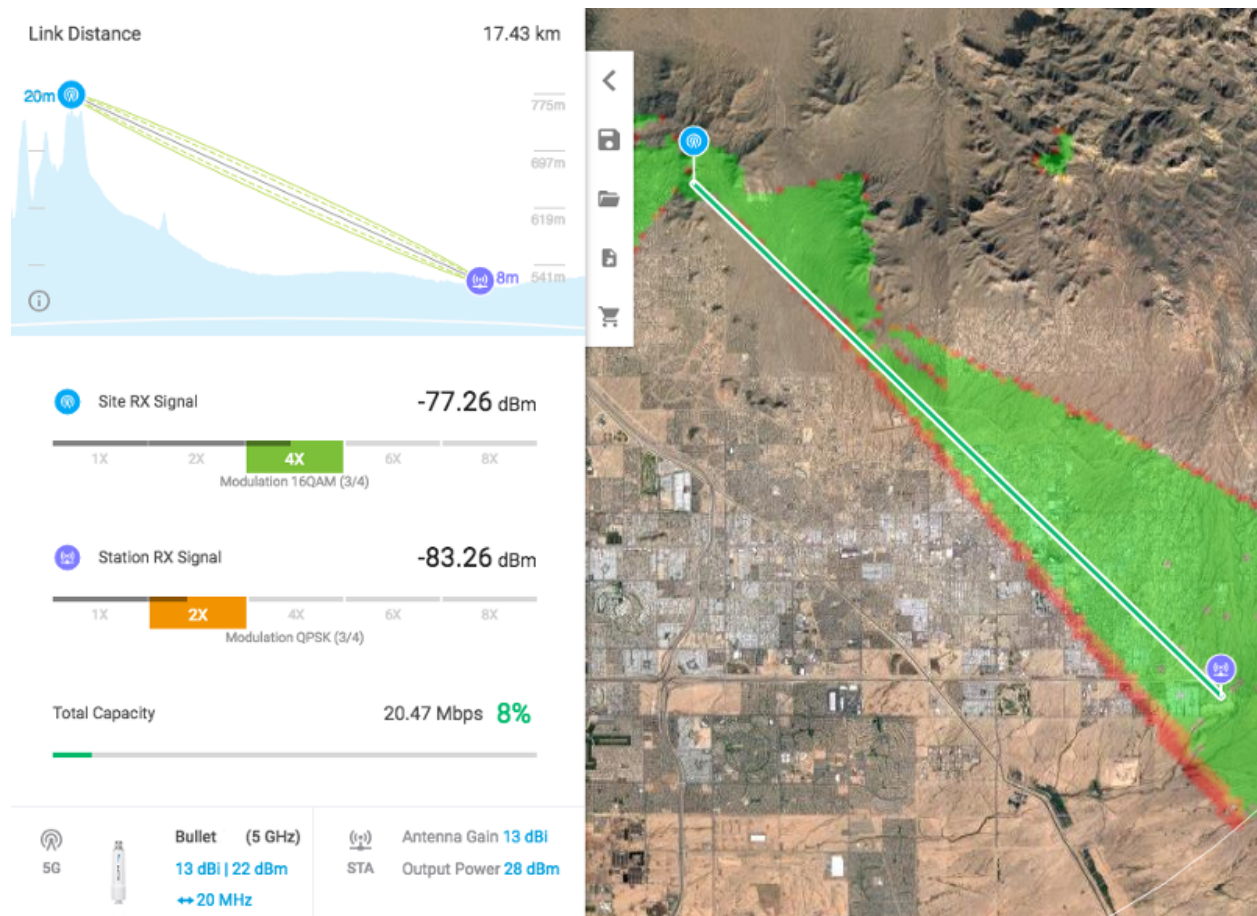
- Antenna Gain
- Receiver Sensitivity

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.

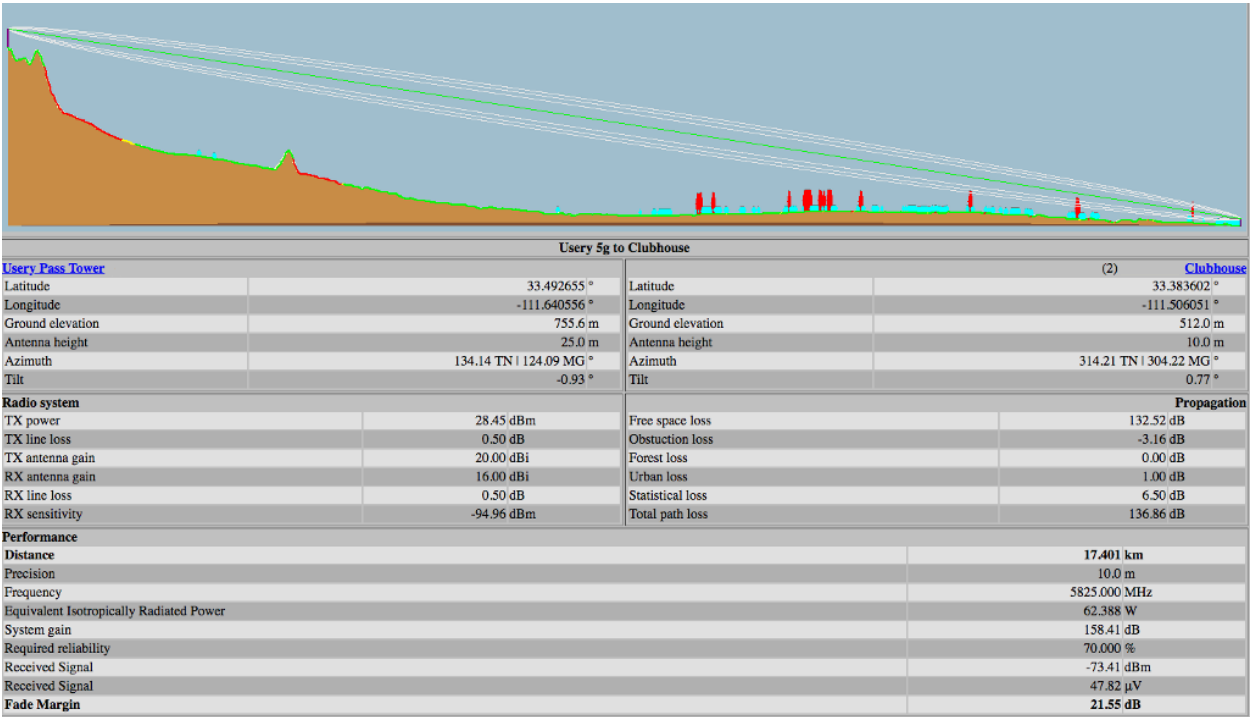
The path profile is color coded to indicate whether the link quality is adequate. It displays the link distance, line of sight, as well as the Fresnel Zone and 60% clearance area. It also estimates the signal levels at each endpoint and the predicted throughput for the link. An example *AirLink* path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

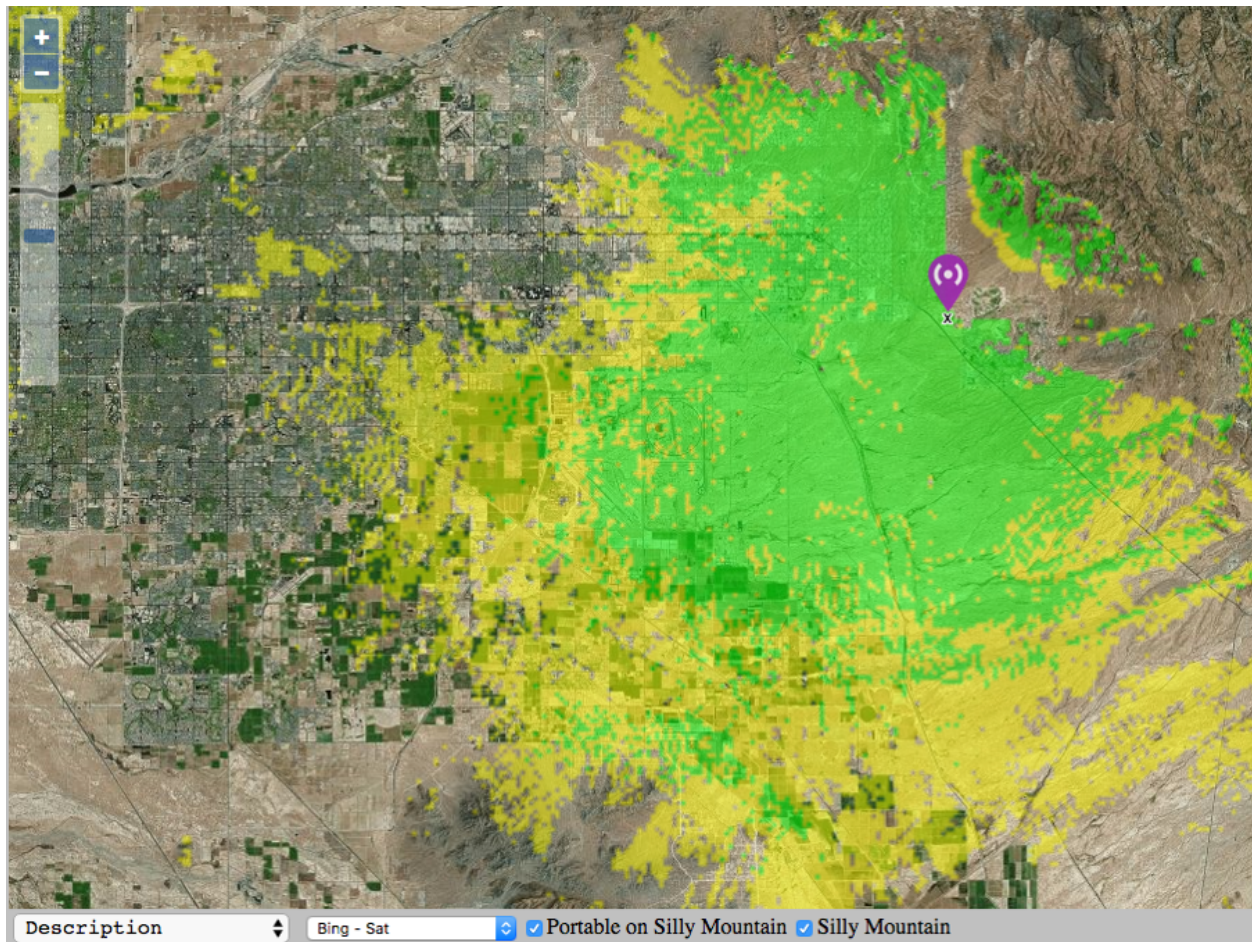


13.2 Determining Node or Network Coverage

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.



CHAPTER 14

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 the types of services that might be useful for communication across a mesh network. Almost any program that can operate across 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.
- 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.”
- As a general rule services should be run on separate LAN-connected computers rather than on the AREDN® node itself. Radio nodes have very limited resources which should be

conserved for node operation rather than 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 are required on your network, and then you will be able to select the best program for the job. Always try to use a program that will cause the least performance impact to your network as it is working to fulfill your communication needs.

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 will operate on a mesh network as long as the server is reachable on a readily accessible network segment with adequate bandwidth.

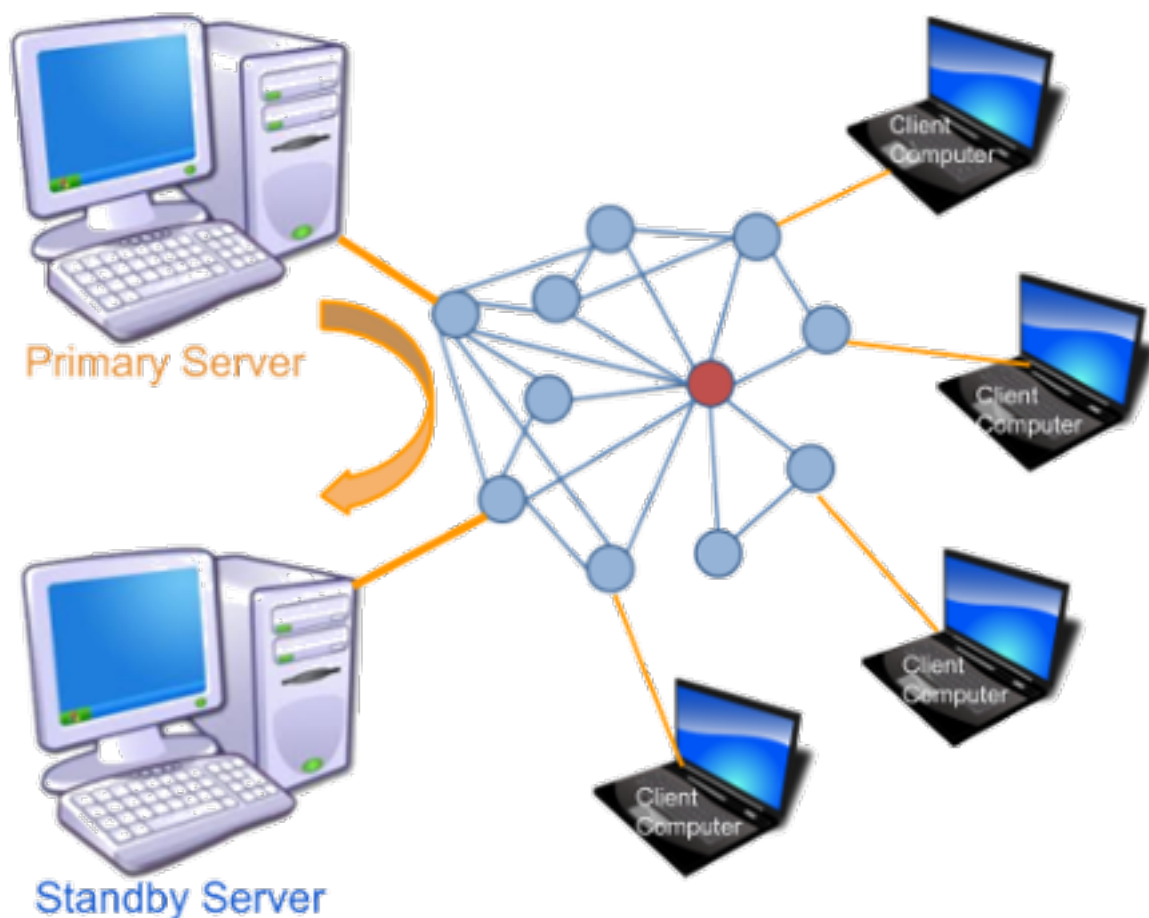
Keeping Multiple Servers in Sync

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.

There are also programs which 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 section 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.

CHAPTER 15

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 specifically for mesh communication. Users access MeshChat via web browser, and the service runs on the mesh node itself or on a LAN-connected Raspberry Pi computer. After logging in by entering a call sign, 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.

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 catch up 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 limited to 500 kb, 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. For additional information about MeshChat, visit this link: [MeshChat](#)

CHATFILESSTATUS

LOGOUT

Mesh Chat v1.0

Zone: MeshChat

Call Sign: KG6WX C

Node: ai6bx-2-chatpi

Updated: 14 seconds ago

Send a Message

New Message

Enter message here

Channel:

Everything

Mesh Chat Users

1

Call Sign	Node	Last Seen
KG6WX C	ai6bx-2-chatpi	1/23/19 10:20 AM

Messages

Enter search

Everything

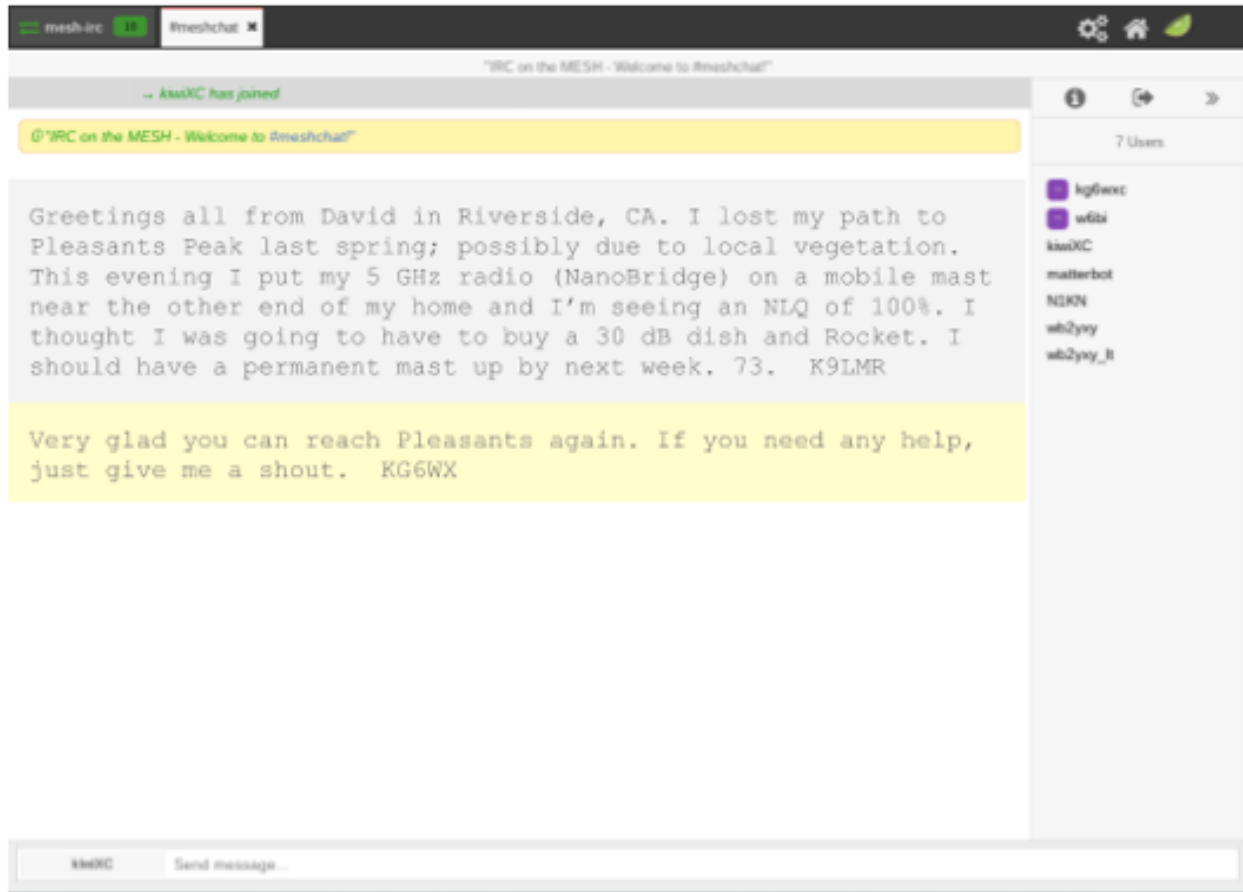
Time	Message	Call Sign	Channel	Node
1/16/19 7:13 PM	Greetings all from David in Riverside, CA. I lost my path to Pleasants Peak last spring; possibly due to local vegetation. This evening I put my 5 GHz radio (NanoBridge) on a mobile mast near the other end of my home and I'm seeing an NLQ of 100%. I thought I was going to have to buy a 30 db dish and a Rocket. I should have a permanent mast up by next week. 73.	K9LMR		ai6bx-2-chatpi

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.

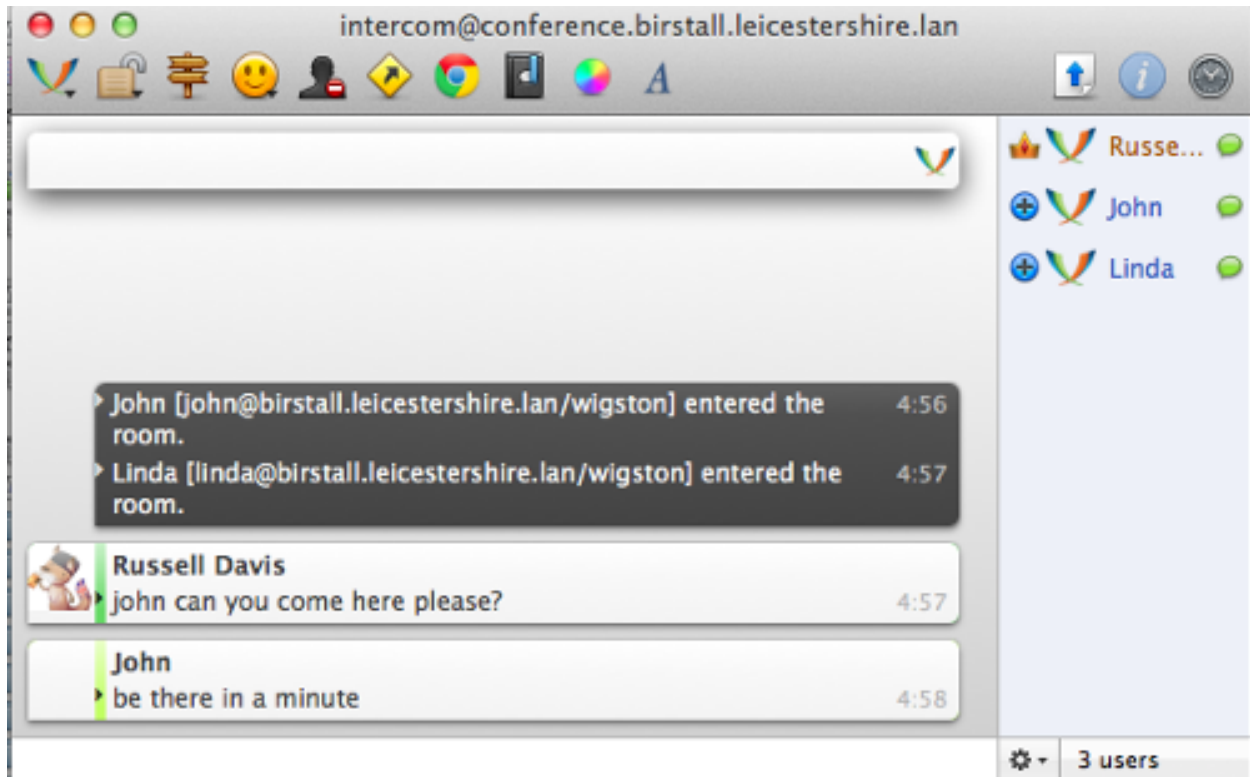
A wide variety of features and functions are available with these and similar chat programs, including various zones, channel types, and user roles. For additional information about IRC services, visit these links: [IRC Servers](#) and [IRC Clients](#)



15.3 Jabber/XMPP

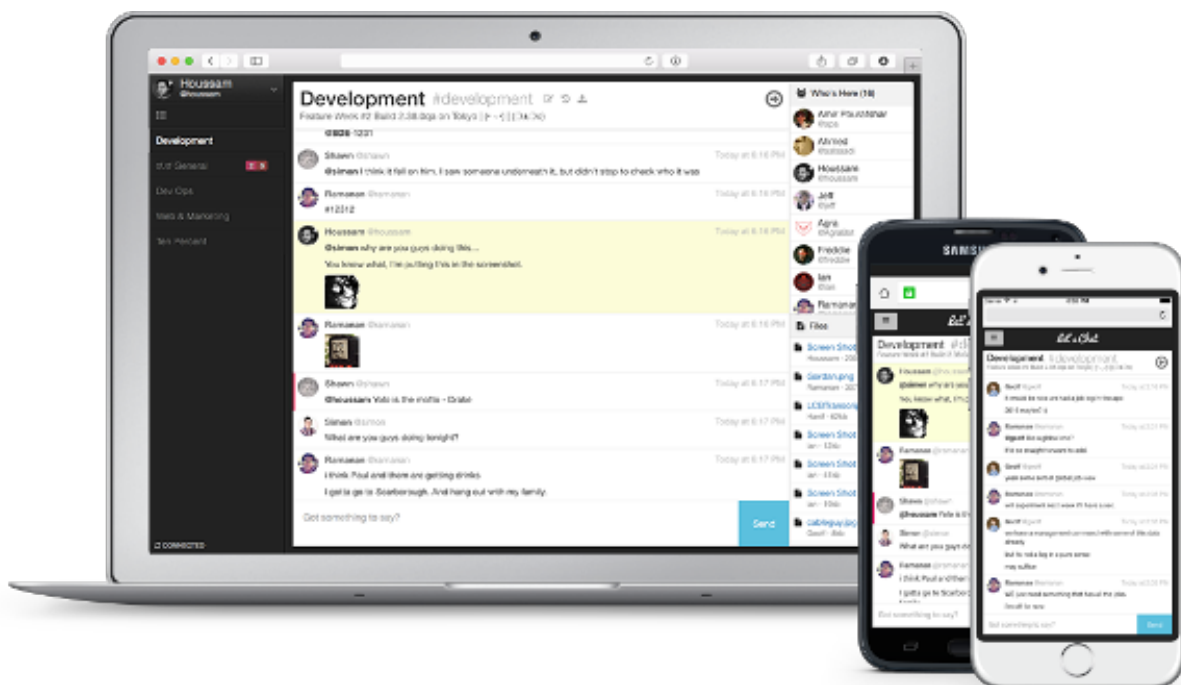
Originally known as Jabber, [XMPP](#) servers have been around for a long time but are fully compliant with modern messaging standards thanks to a large community of developers worldwide. These servers provide one-to-one messaging as well as group chat sessions. User lists have activity and presence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

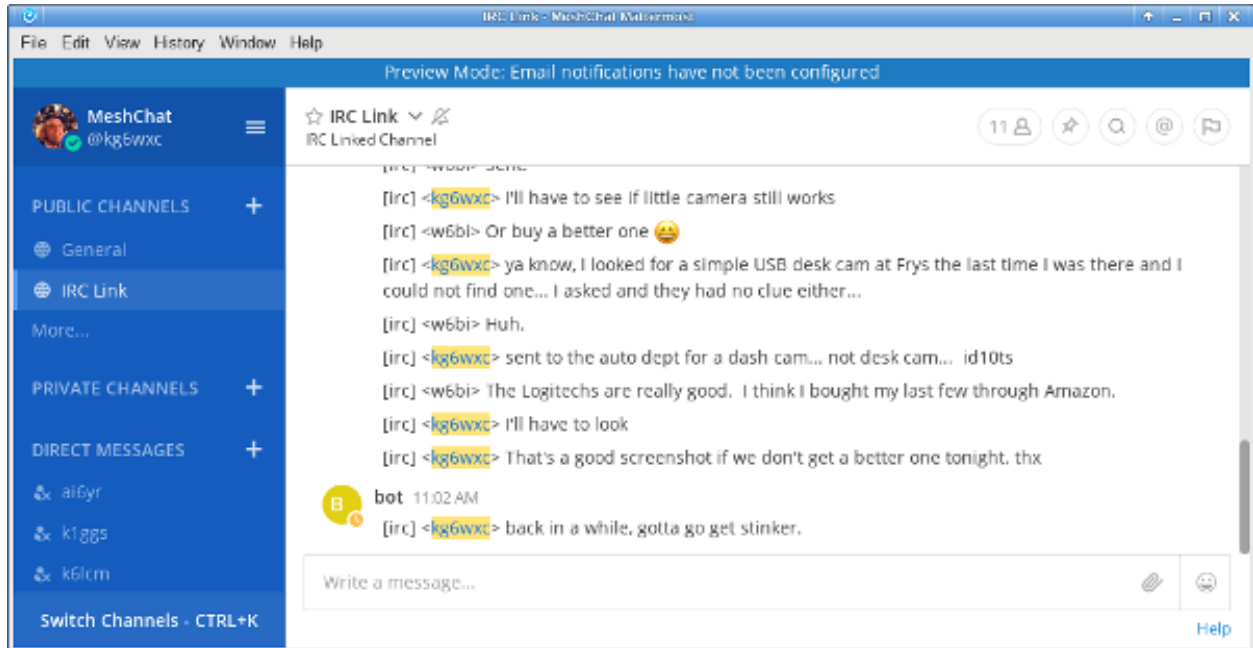
Let's Chat is an open source messaging service for small teams. It provides one-to-one communication between [XMPP](#) users as well as group messaging and @mentions in a variety of chat rooms. Searchable conversation history is available, in addition to text and image pasting, user activity notifications, 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.

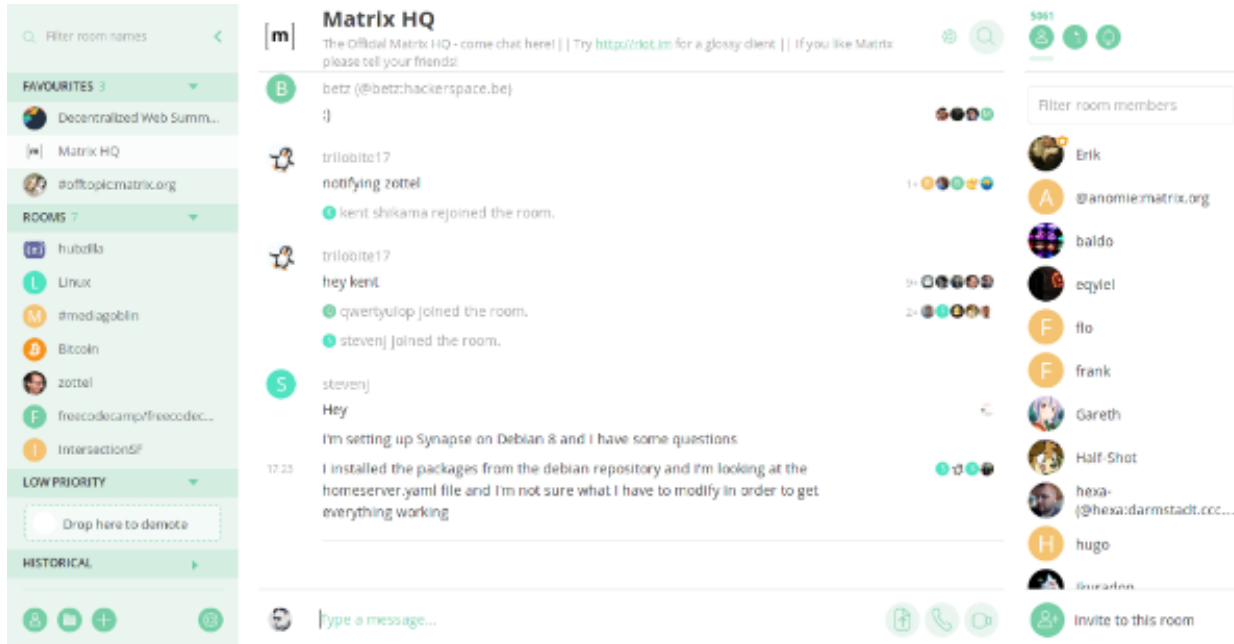
Mattermost supports @mentions, and channels are available for organizing conversations which can be topic-based, group-based, or event-based. Notifications indicate user presence and activity. File sharing is provided for PDF and text files, as well as audio, video, and image files. For additional information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

CHAPTER 16

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 complicated 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.

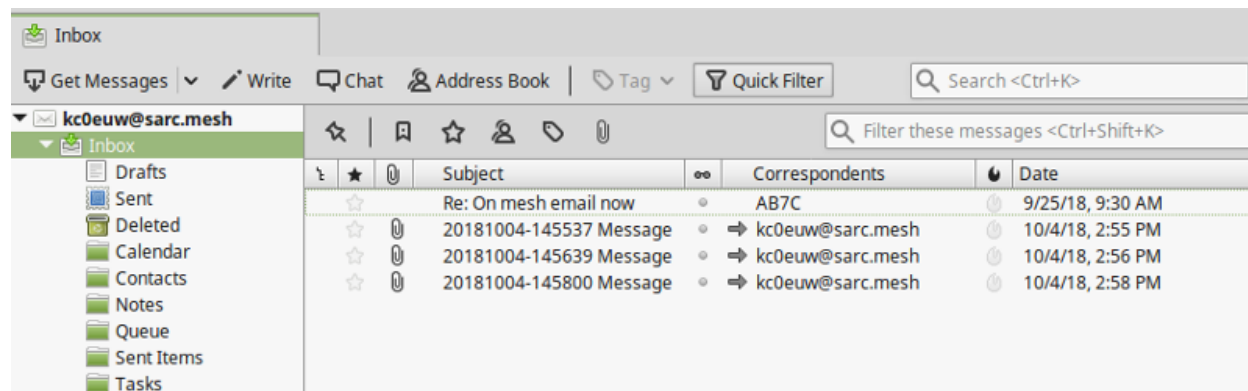
SMTP In order to implement an email server you will need to select a software package to handle the Simple Mail Transfer Protocol. You can select one of the example open source packages from the list below, or you can implement another SMTP agent of your choice.

- [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.

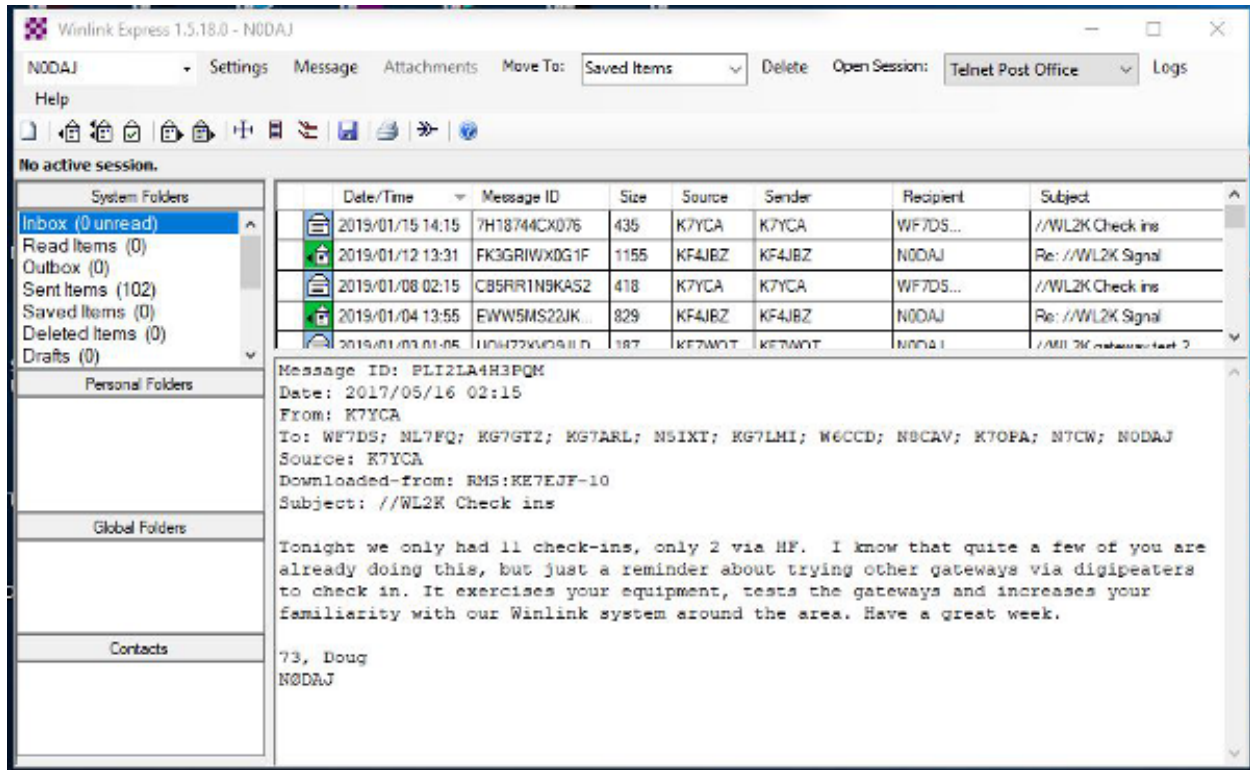
You will need to have detailed knowledge and skills when building your own open source email server, with the advantage of having complete control over everything on the system. There is some administrative overhead for creating and maintaining all user email accounts as well as handling other management 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 *RMS 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). Refer to the information link below for details about the specific network port settings that will be required. The maximum attachment size is currently 5MB per message as compared to the 100KB limitation on HF and

Packet RMS stations. For additional information, please visit the AREDN® forum category on Winlink located here: [Winlink Forum](#)



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

CHAPTER 17

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.

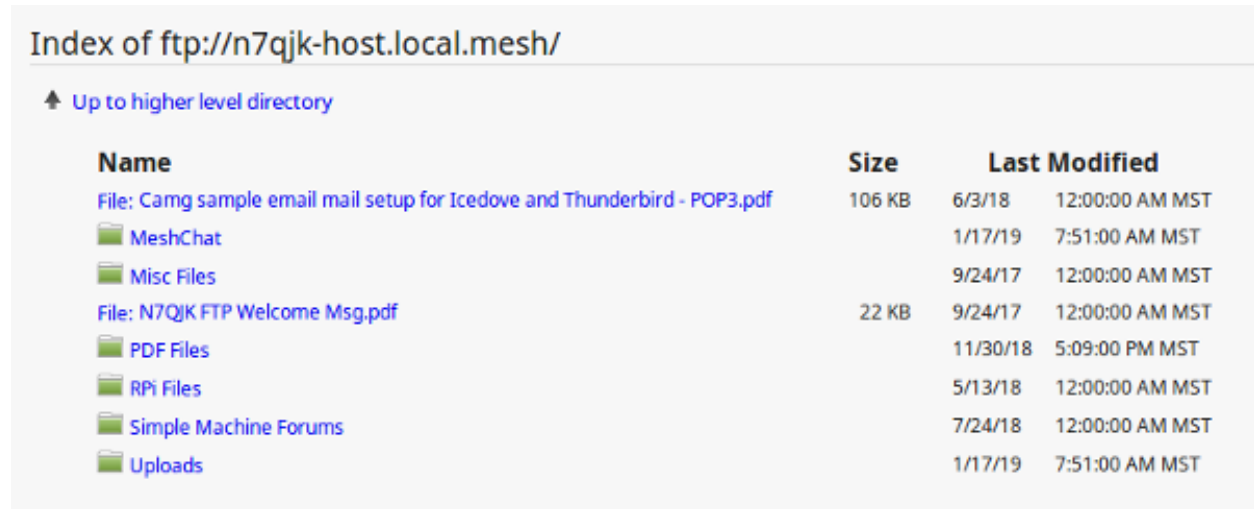
File transfer protocols themselves have minimal impact to network performance, but downloading a very large file across a mesh network could have a major performance impact. Transferring text files, and especially compressed text, should have minimal impact to the network, but a network could experience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



Index of <ftp://n7qjk-host.local.mesh/>

[↑ Up to higher level directory](#)

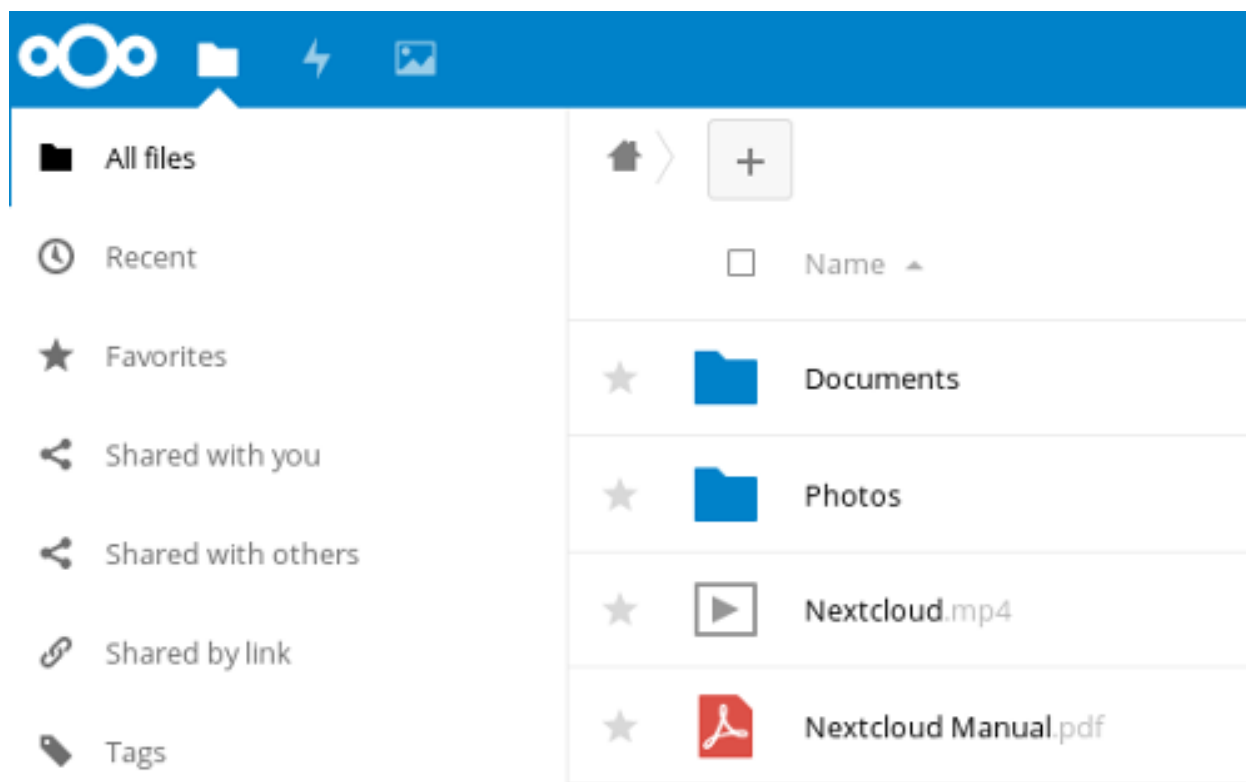
Name	Size	Last Modified
File: Camg sample email mail setup for Icedove and Thunderbird - POP3.pdf	106 KB	6/3/18 12:00:00 AM MST
MeshChat		1/17/19 7:51:00 AM MST
Misc Files		9/24/17 12:00:00 AM MST
File: N7QJK FTP Welcome Msg.pdf	22 KB	9/24/17 12:00:00 AM MST
PDF Files		11/30/18 5:09:00 PM MST
RPi Files		5/13/18 12:00:00 AM MST
Simple Machine Forums		7/24/18 12:00:00 AM MST
Uploads		1/17/19 7:51:00 AM MST

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.

AREDN Help File

Please note:

- Clicking the AREDN logo will redirect to <http://localnode.local/mesh>
- Javascript and page redirection must be enabled in your browser for the web interface to work.
- Some operations can take several seconds, or even longer, to complete. There is currently no feedback while the node is working on your request. Be patient and wait for the web interface to respond before trying to click other buttons.
- Avoid the use of your browser's back, forward, and reload buttons. Every page has navigation controls to take you where you want to go.
- The various pages of the web interface are intended to be used by only one person at a time. This is especially important on the setup pages where using them from multiple browsers or multiple computers at the same time will almost certainly cause problems. Viewing different pages at the same time should not cause any conflicts.

Status Page

This is the first page you will see when accessing <http://localnode/> or <http://your-node-name/>. The top bar displays the node name and also a tactical name if one has been assigned. For more about tactical names see the Basic Setup section. Below the name bar there will be a few control buttons. Some of these buttons may not be available depending on the current configuration:

- Refresh** will update the page with current data.
- Mesh Status** takes you to a page which shows what Neighbor nodes and Remote nodes are visible as well as what services are being provided through those nodes.
- OLSR Status** takes you to the web pages that OLSR itself provides which gives you detailed information about the current state of the OLSR routing software.
- WiFi Scan** displays a list of other 802.11 signals that the node can see and only of the same bandw. 802.11 signals include Access Points (AP), neighbor nodes (connected ad-hoc stations), and other networks (foreign ad-hoc networks). The AREDN mesh is created on top of an 802.11 'ad-hoc' network. Consequently when multiple ad-hoc networks are visible to each other (different SSID or channel) is displayed and not individual nodes (stations). There is also an automatic scan mode. It is not recommended to run a wifi scan continuously because this will degrade mesh performance. A wifi scan transmits query channels to discover other devices.
- Setup** takes you to the setup pages of the web interface. You will need to supply a username and password to access those pages. The username is always "root", and the password is the one you set on the Basic Setup page. If the node has not yet been configured, the password is "hsmm". Note that the password given for the setup pages is NOT encrypted in transit.
- Select Theme** switches display themes/styles. Black on white was chosen because it provides the

Chat

KCOEUW: hello everyone 12:53

CHAPTER 18

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 *meshnet*. 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 packages. Asterisk is extremely robust tried-and-true IP-PBX software, but you will need specific knowledge and skills to implement it.



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

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



18.2 VoIP Endpoints

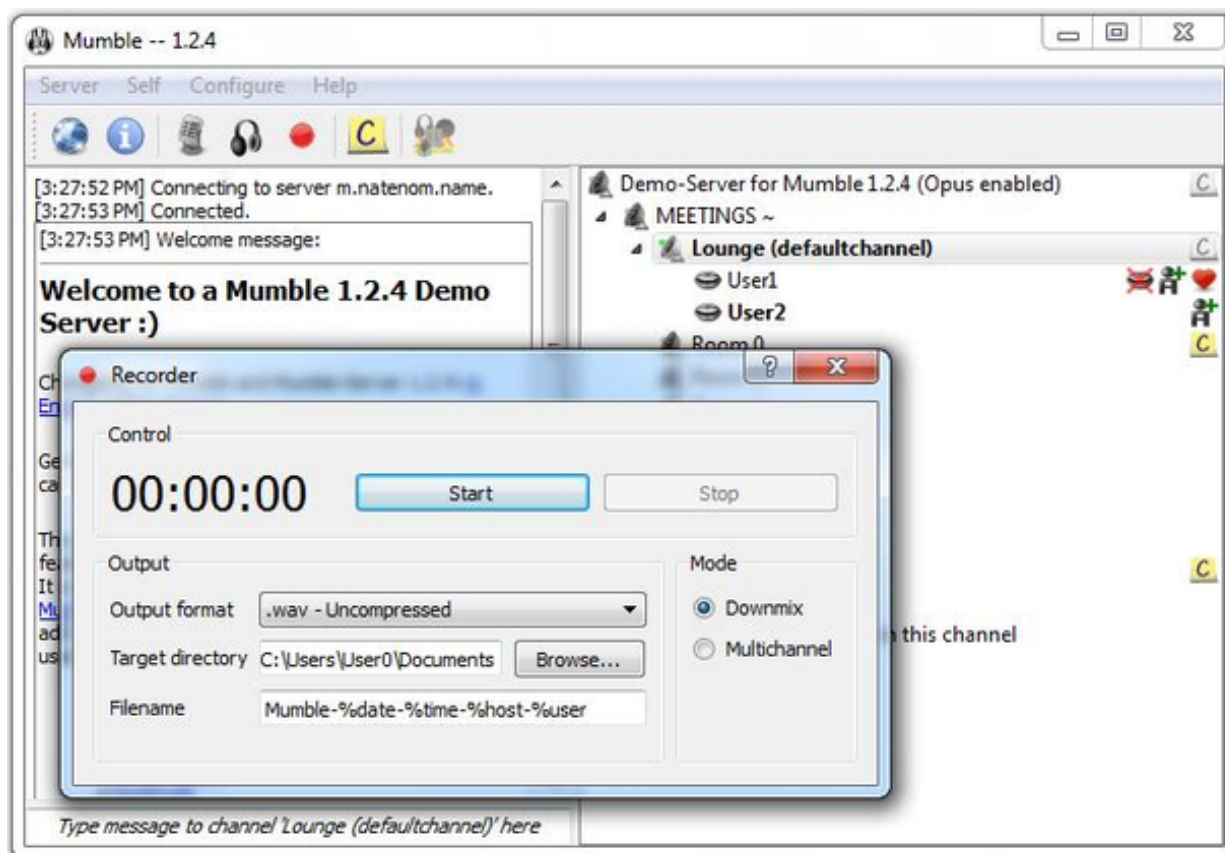
Once you have a VoIP PBX provisioned on your mesh network, you will need VoIP endpoints which can communicate through the server. Specialized [VoIP phone](#) hardware is available from several manufacturers which can provide communication endpoints on your network. It is also possible to use legacy analog 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.

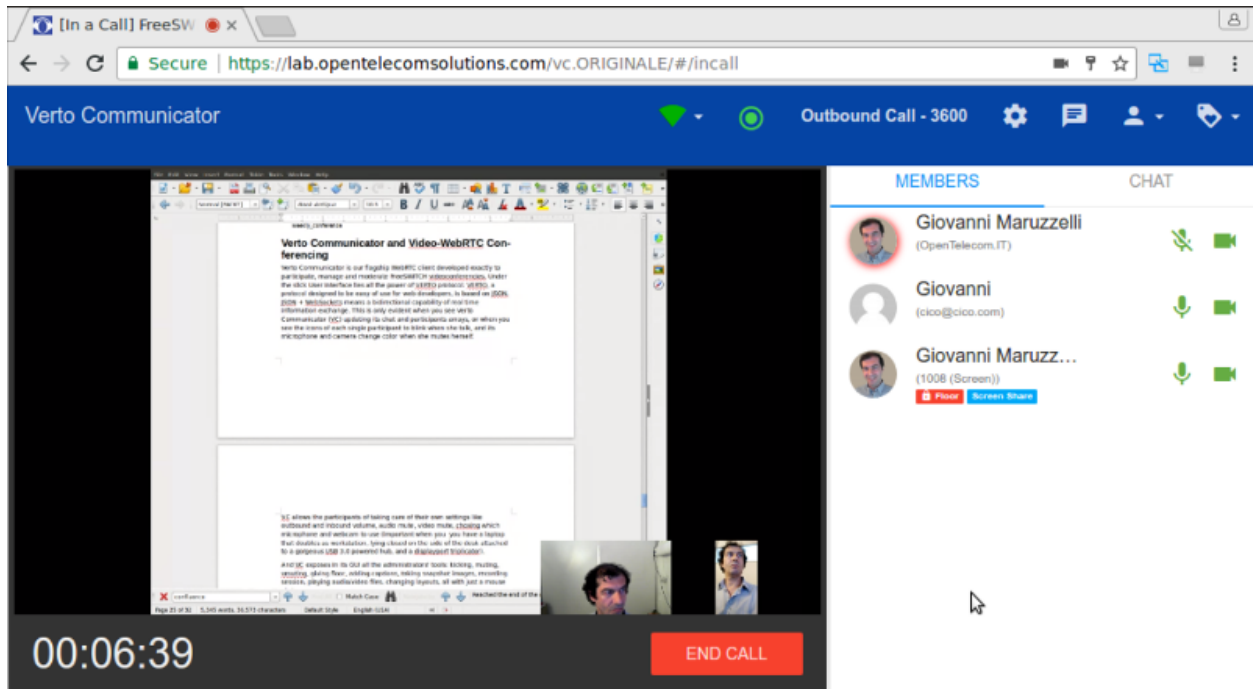
Hosting Mumble locally requires downloading the *Murmur* server, which is included as an option in the Mumble installer. The primary users of Mumble are Internet video gamers who want to communicate with each other during game play. However, it can also be used as a non-gaming voice communication service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



18.3 Video Conferencing Software

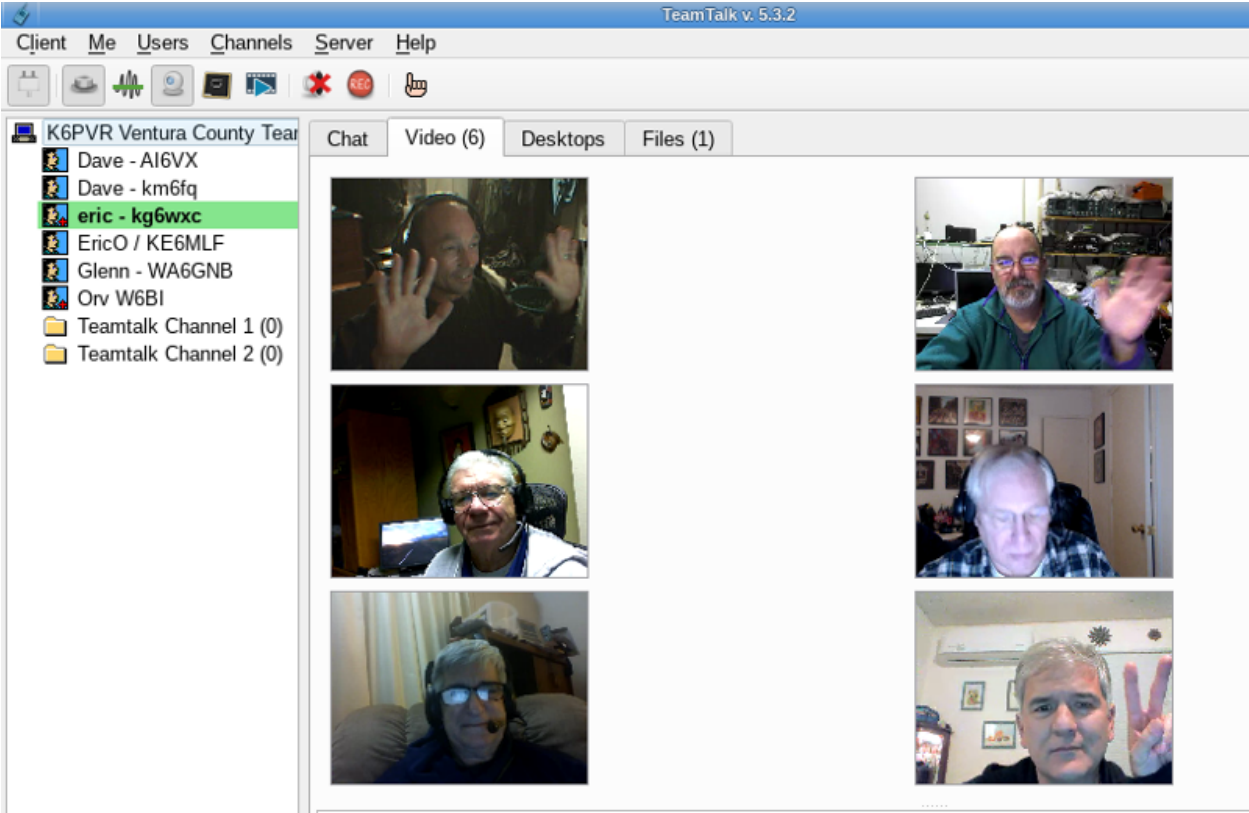
FreeSWITCH Server [FreeSWITCH](#) is a recent communication platform that can be used to build voice PBX systems with voice response menus, video conferencing with chat messaging and screen sharing capabilities, and full [WebRTC](#) support. Its modular design makes it possible to install only what is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

CHAPTER 19

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](http://ispyconnect.com), 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. There is a *Cameras* forum topic on the AREDN® website where you can post your questions and experiences: [arednmesh.org camera forum](https://arednmesh.org/camera-forum).

19.2 Video Display Software

The software described in this section can help you to provision video surveillance services on your *meshnet*. 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.

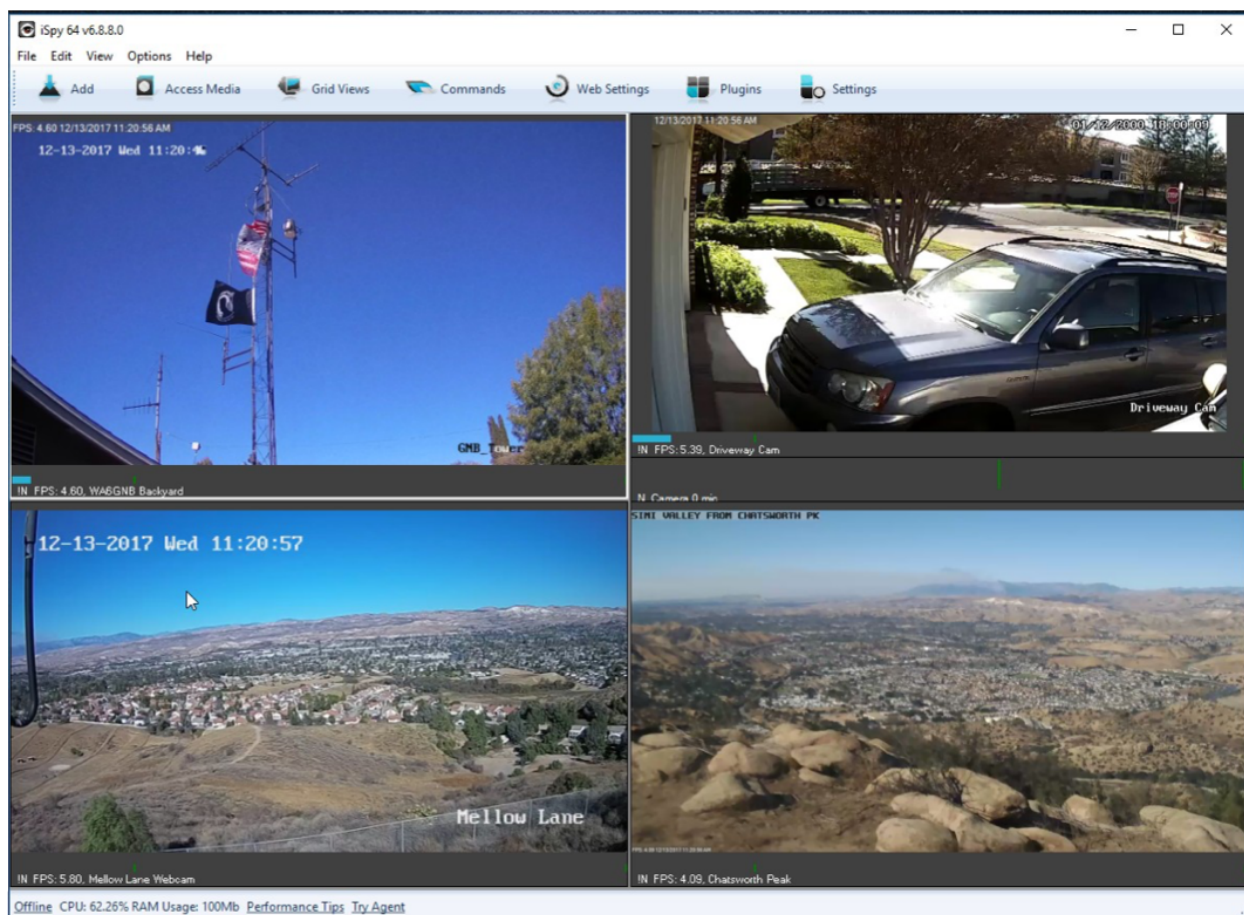
19.2.1 iSpy

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`.

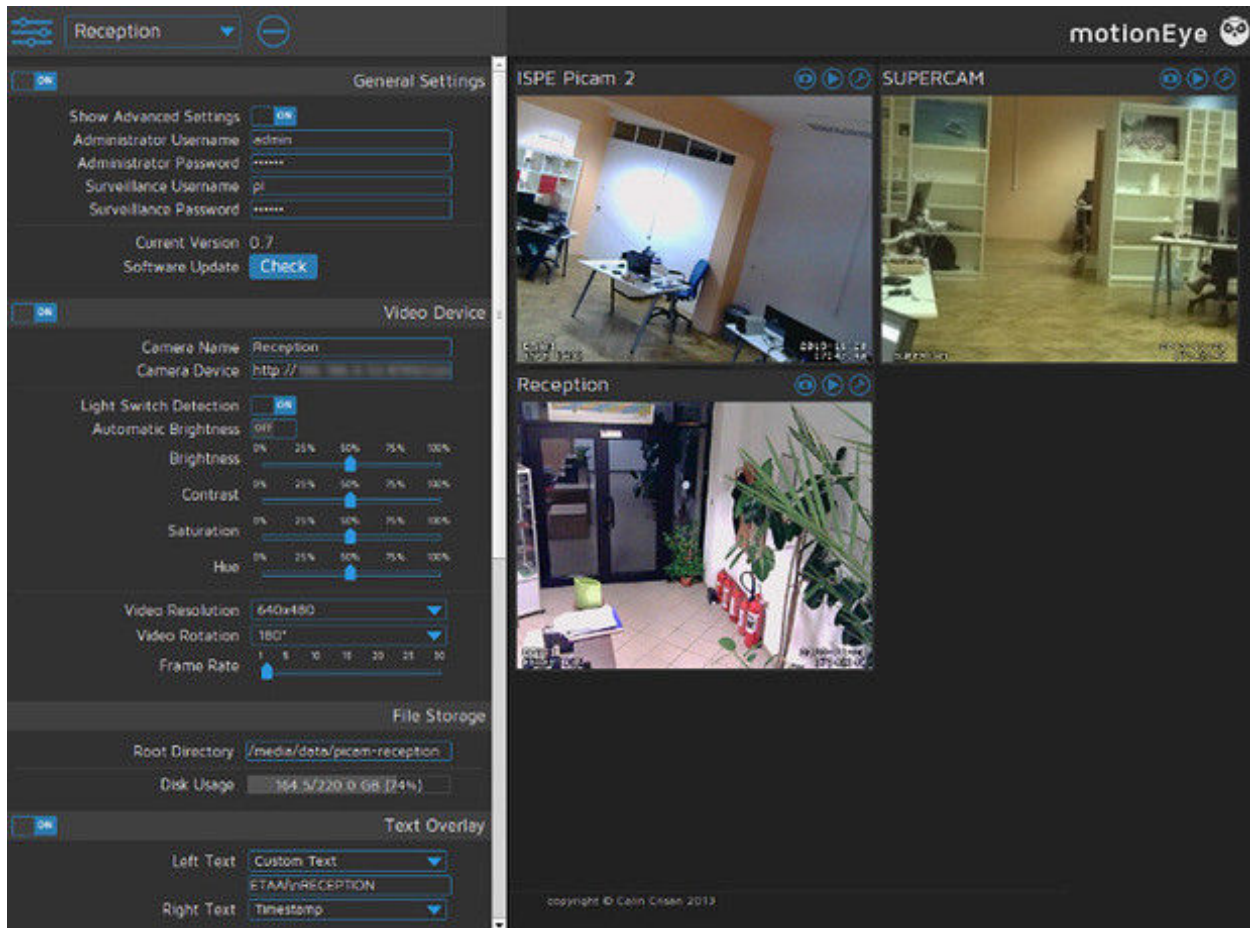
In the lower right video stream on the iSpy display below you can see the smoke plume from the 2017 [Thomas Fire](#) in California, which was recorded by a camera on the local AREDN® network. For additional information about iSpy, visit this link: [iSpy](#).



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](#)

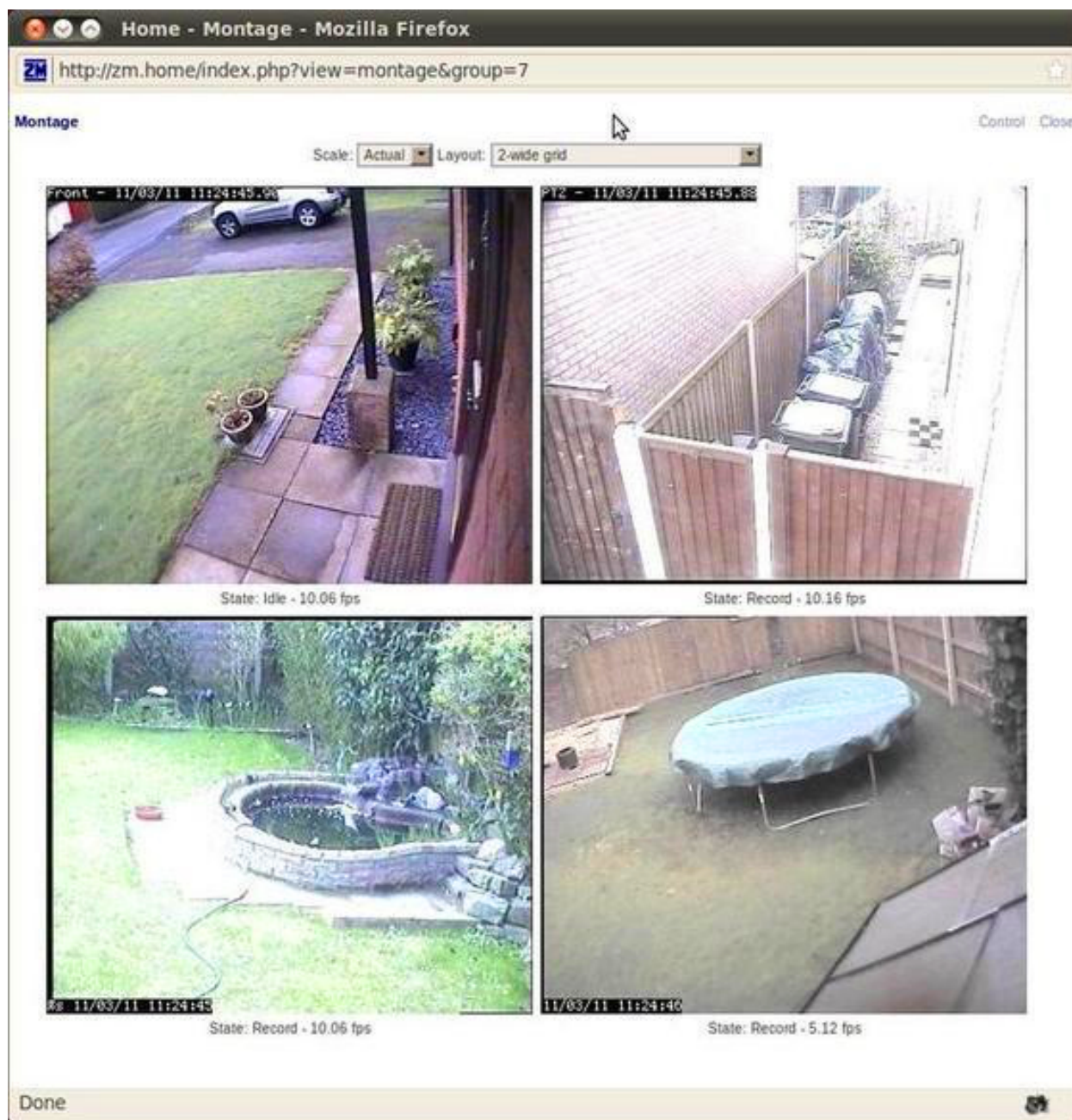


19.2.3 ZoneMinder

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](#).



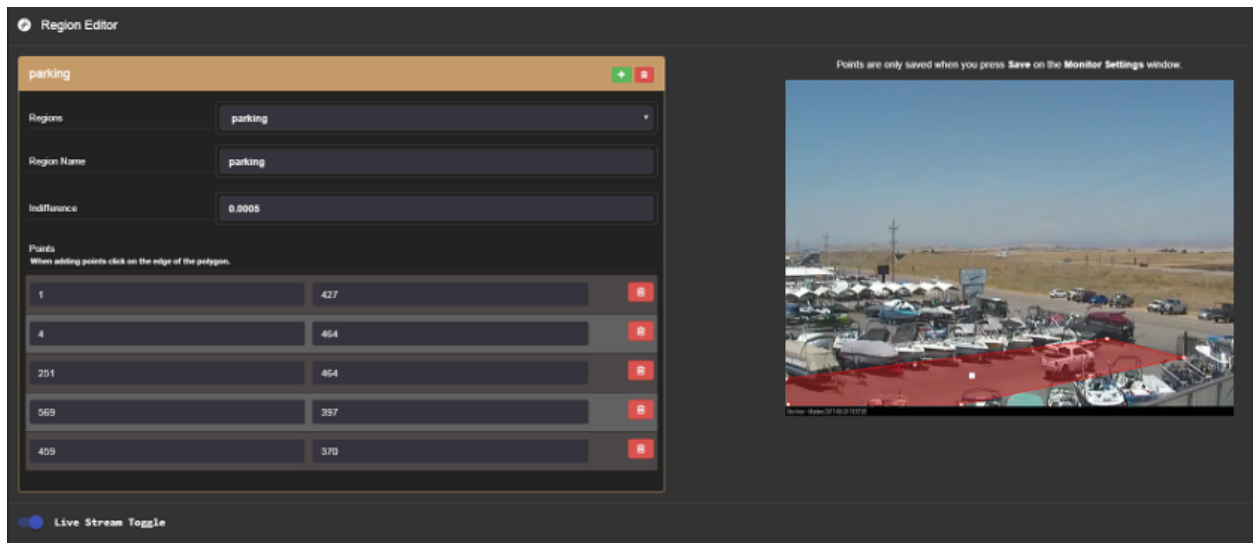
19.2.4 Shinobi

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 cam-

era 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 <i>NC</i> use	medium	lin/mac	medium

NC ~ non-commercial

CHAPTER 20

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.

The screenshot displays the EmComMap v0.4a web interface. At the top, it shows the title "LA Example" and "Sample incident for experimentation" in red, along with a "TESTING" status. The user is logged in as "k6oat". The main map area shows a geographic region with various icons representing resources. The right sidebar contains tabs for "Traffic", "Operators", "Locations", and "Incident". The "Traffic" tab is active, showing a table of messages. The table has columns for "From", "To", "Time", "Rel. time", "Location", "Prec.", and "Attachment". The messages listed are test messages from k6oat to k6da. The bottom of the sidebar has a form to submit a new traffic message.

From	To	Time	Rel. time	Location	Prec.	Attachment
k6oat		2018-12-08 16:54	1 minute ago	CHH	E	
TESTING: Power failure reported at Hollywood area hospitals						
k6oat		2018-12-08 16:54	1 minute ago	CHH	P	
TESTING: Bad traffic in Hollywood. Avoid if possible.						
k6oat	kk6da	2018-12-08 16:53	2 minutes ago	CHH	R	
TESTING: En route to CHH						

From: k6oat To: Related location: CHH Precedence: Emergency
 Message:
 Attachment: Choose File No file chosen Submit traffic

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](#).

Tickets 3.30A Beta on sarcmesh.org Logged in: admin : Super Module: add Time: 07:17 Day Night

Situation New Units Fac's Search Reports Config SOP's Chat Help Log Full scr Personnel Links Board Mobile

Current situation - Your area Viewing Regions (mouse over to view) Normal 1, Medium 0, High 0 Page Loaded

Incidents
click item to view / edit, right click for act / pat / notes, Click headers to sort

Icon	Scope	Address	Type	A	P	U	Updated
1	2/Flash flood	411 East Scenic ..	examp1	0	05	07:16	

Responders
click on item to view / edit, Click headers to sort

Icon	Handle	Name	Mail	Incidents	Status	M	As of
1	KC0EUW	Steve			available		05 07:04

Facilities
click on item to view / edit, Click headers to sort

Icon	Name	Mail	Status	Updated
2	Fire Station 22		Open	05 07:10

Map
Show Assigned
Road Conditions
Contact Units
Contact Facilities

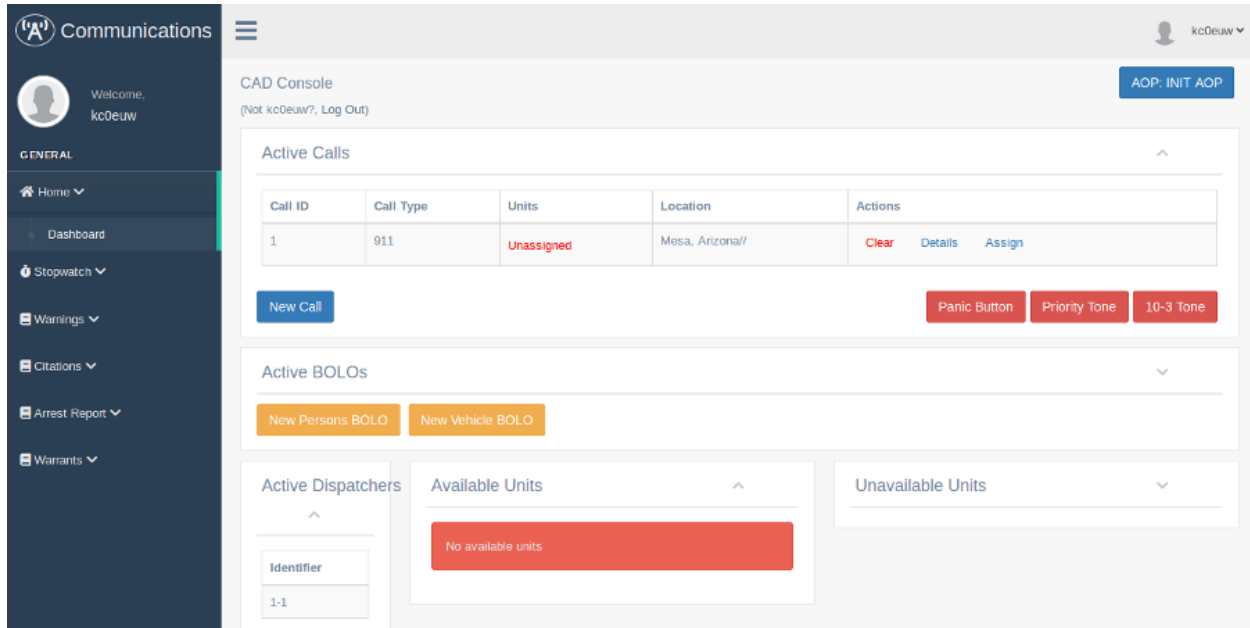
Map data © 2011 OpenStreetMap contributors, Leaflet

20.3 OpenCAD

Like *ISES Tickets* described above, *OpenCAD* is a web server application which can run on any computing platform that supports a traditional LAMP stack. *OpenCAD*, however, is not map-based and does not provide GIS mapping features. It is aimed primarily at creating and tracking calls in a law enforcement context. Several user roles are defined, each with access to specific dashboard views tailored to their responsibilities. These roles include communications/dispatch, police, fire, EMS, sheriff, highway patrol, roadside assistance, and civilian. The main task of

OpenCAD administrators is to approve new user access requests and to manage user settings across the system.

Users with law enforcement roles can view BOLOs (Be On the Look Out) and active calls, as well as creating citations, warnings, and arrest reports. Users with fire and EMS roles can view and edit call details, as well as accepting call assignments. Dispatchers can create, edit, and assign calls, track resource availability, as well as viewing BOLOs, citations, warnings, arrest reports, and warrants. Civilian and Roadside Assistance users can create calls. For additional information about *OpenCAD*, visit this link: [OpenCAD](#).



There is an older package similar to *OpenCAD*, but with fewer features, called *ampCAD*. Information is available here: [ampCAD](#)

20.4 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
OpenCAD	open source	small	win/lin/mac/rpi	medium

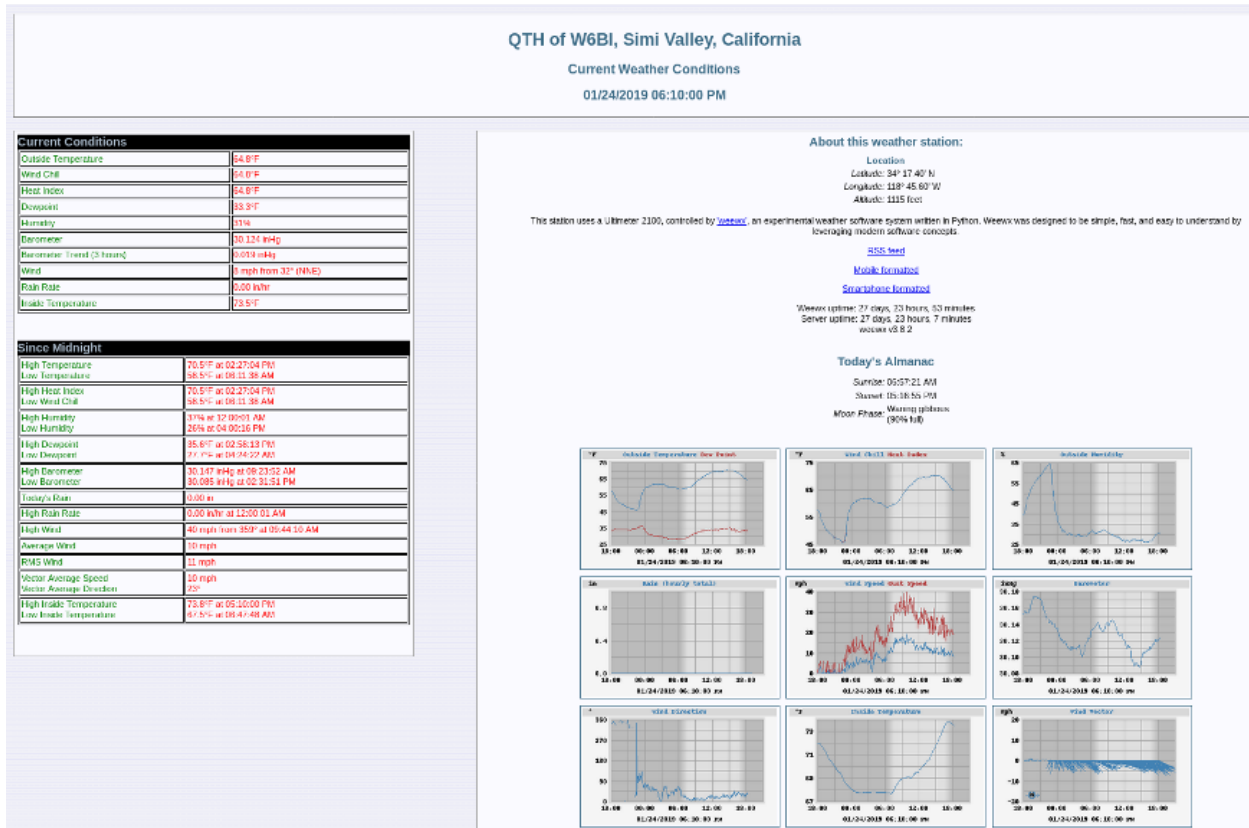
CHAPTER 21

Other Possible 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 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](#).



21.2 Network Time Services

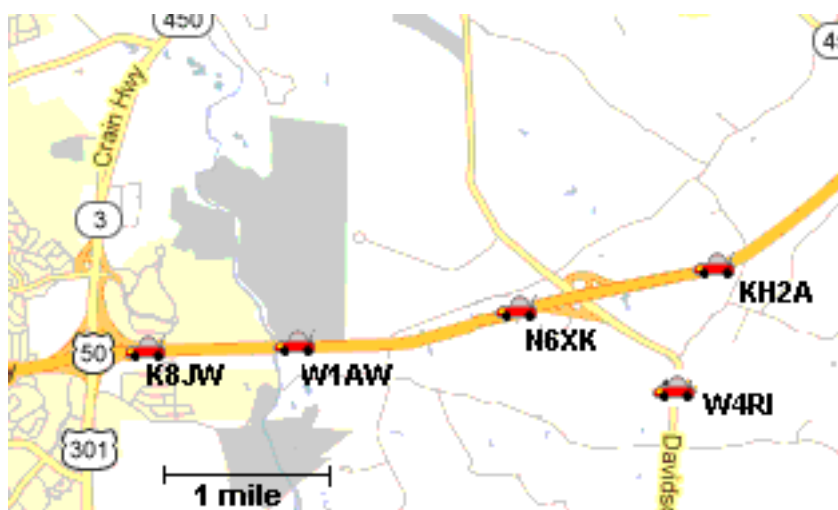
Although the AREDN® nodes themselves do not depend on network time synchronization, there may be other programs or services running on your mesh network which would benefit from having accurate network time updates. **Network Time Protocol (NTP)** is a reliable way for networked devices to update their 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 post](#).

21.3 GPS Tracking Services

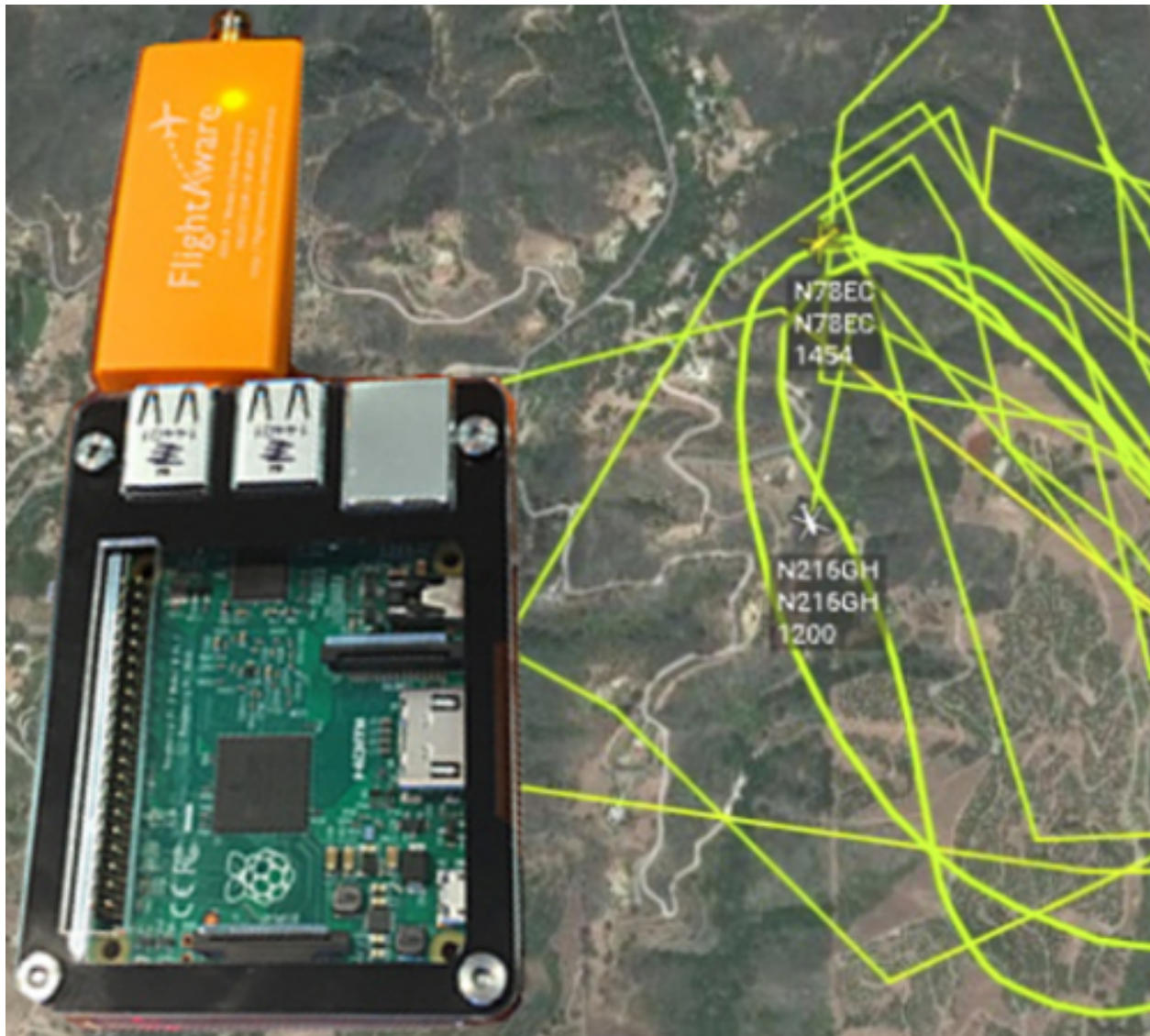


Tracking deployed resources is an important task during any emergency. There are many options for monitoring and displaying the GPS locations of tracked resources, two of which are mentioned here.

Many amateur radios and portable locating beacons transmit [Automatic Packet Reporting System](#)

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

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



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

the [AREDN Forums](#) for additional information, ideas, and how-to posts about possible services for mesh networking.

CHAPTER 22

Firmware Upgrade Tips

Upgrading an AREDN® node is a straightforward process 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. The newest firmware versions have a built-in check to verify that the firmware image you selected is appropriate for the device on which you are installing it. Earlier firmware versions (3.16.1.x and 3.18.9.0) do not have these checks, so be sure you selected the correct firmware version for your device before starting the upgrade.

Here are some “best practice” tips to assist with the firmware upgrade process. These ensure that memory utilization is at its minimum on the node. The upgrade process can fail due to lack of memory, but such a failure will leave the node unchanged on its previous firmware version.

Before starting the firmware upgrade, it may be necessary to stop, disable, or uninstall Meshchat, hamchat, snmp, and any active tunnels. 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 will ensure that its RAM utilization is at a minimum.

Use a stepped approach to firmware upgrades. For example, if your node is running version 3.16.1.0 you should probably upgrade to version 3.18.9.0 before attempting to apply a newer version.

CHAPTER 23

How-to Use PuTTYGen on Windows to Make SSH Keys and Use Them on AREDN® Nodes

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.

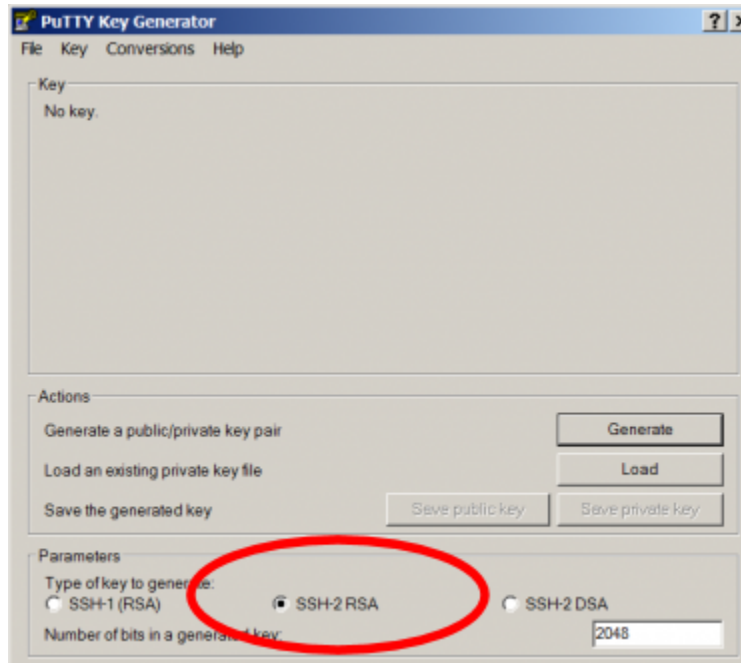
The use of Secure Shell (SSH) keys when using PuTTY or another SSH client is a useful aid to managing a group of AREDN® nodes.

First, obtain the PuTTY suite of applications from the [PuTTY Download Page](#) and install them on your computer.

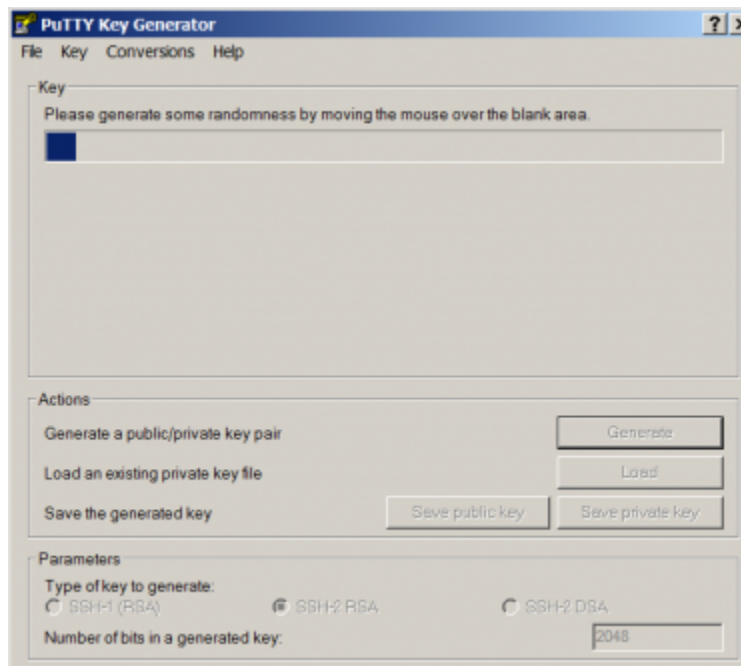
Second, obtain and prepare to use a text editor such as [Notepad++](#) that does not insert unwanted characters and metadata into a text file.

Next, follow the steps below.

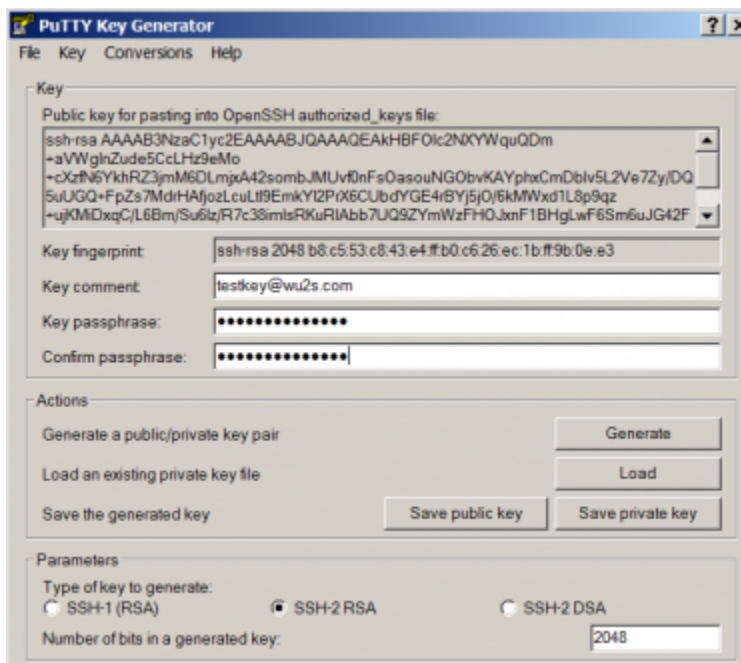
1. Start the PuTTYGen application. Confirm that you are going to generate an SSH-2 RSA key.



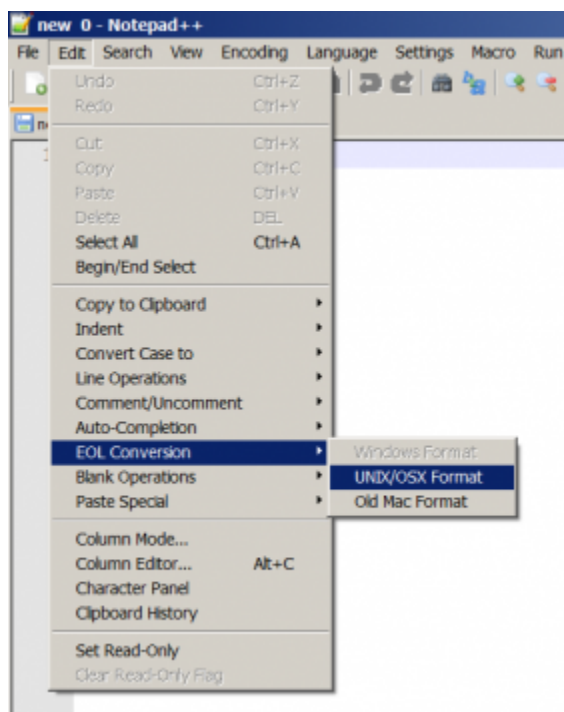
2. Select the *Generate* button to get the prompt asking you to make some random mouse movements. After a short while you get a message asking you to wait while the keys are generated. It finishes and you now have a new key pair.



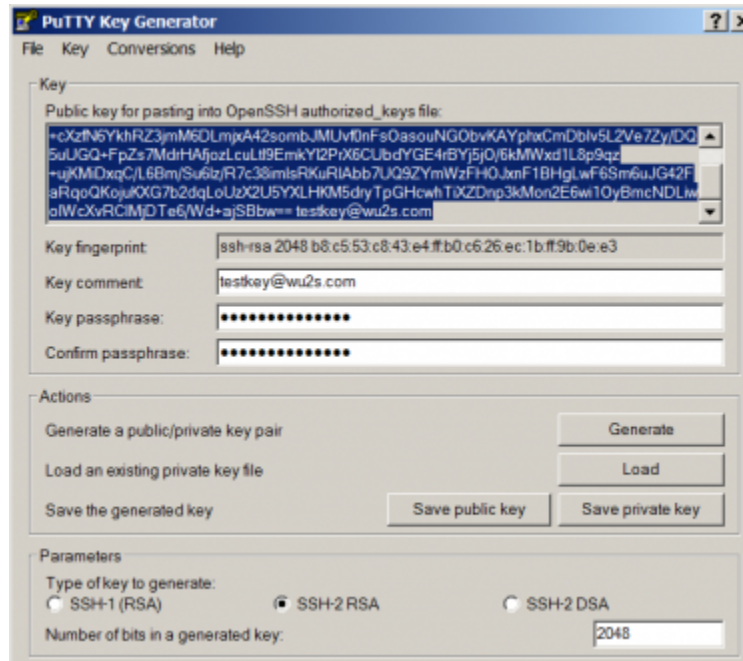
3. Give the key pair a suitable comment so that you will remember what the keys are used for. Here we just entered `testkey@wu2s.com` for an example. Whatever you enter in the “Key Comment” field must look like an email address with no spaces and the “@” present as in *callsign@example.com*. Also enter a suitable passphrase to use when accessing the private key. Record this passphrase so you will remember it for future use.



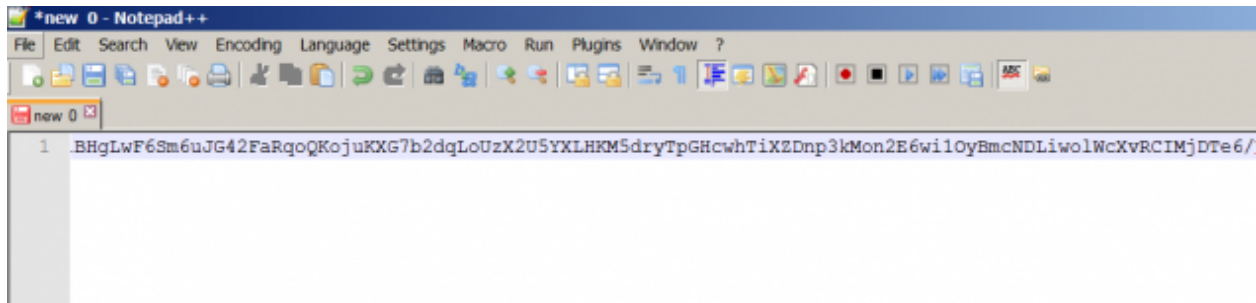
4. Now copy and save the public key. Open Notepad++ and confirm that the End Of Line (EOL) format is set to UNIX/OSX Format. This will ensure that there are no extraneous characters in the public key file.



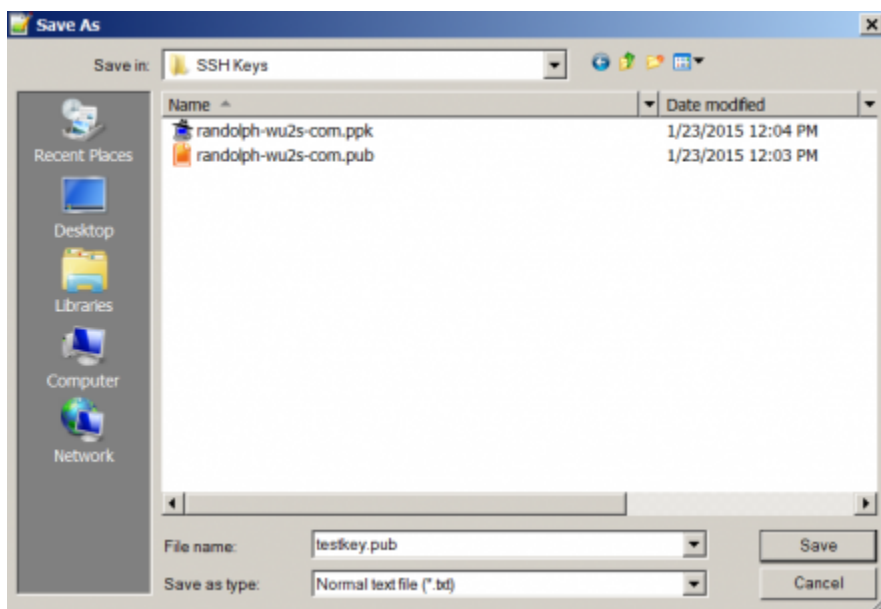
5. Back in your PuTTYGen window, select and copy (Control-C) the complete text in the boxed labeled “Public key for pasting into OpenSSH authorized keys file”



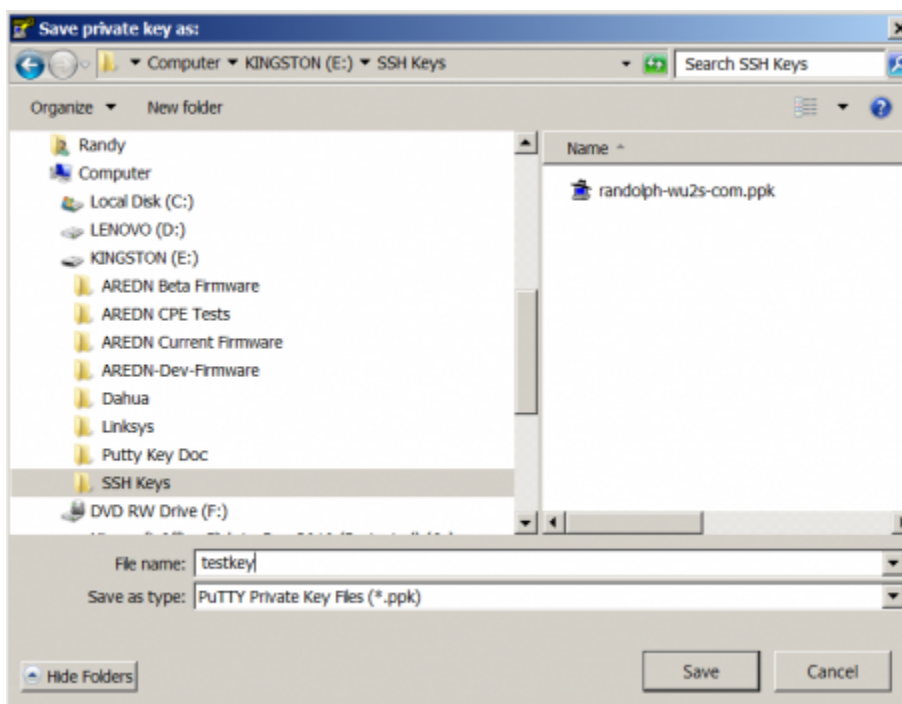
6. Switch back to your Notepad++ window and Paste (Control-V) the public key text you just copied from PuTTYGen.



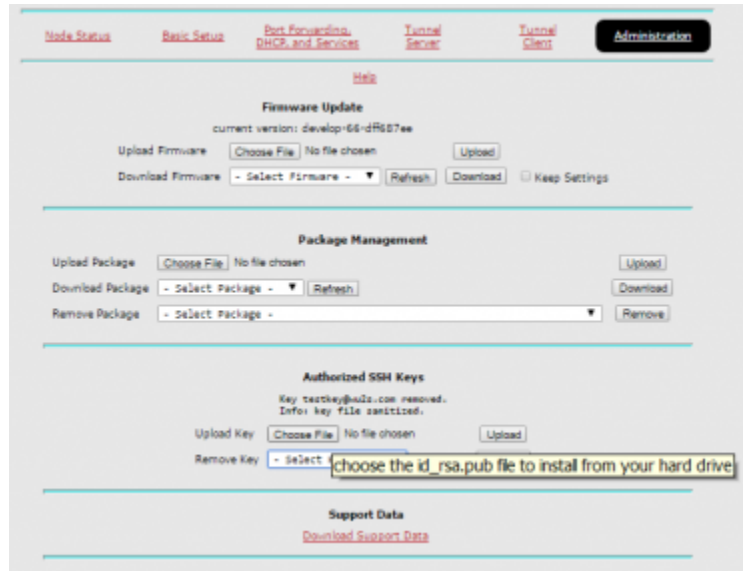
7. From the Notepad++ menu bar, select File -> Save As to save the public key to a suitable location. Many people save their keys on a USB flash drive to maintain physical possession of them at all times. Give the public key file a suitable name. You can exit Notepad++ now since you will not need it again.



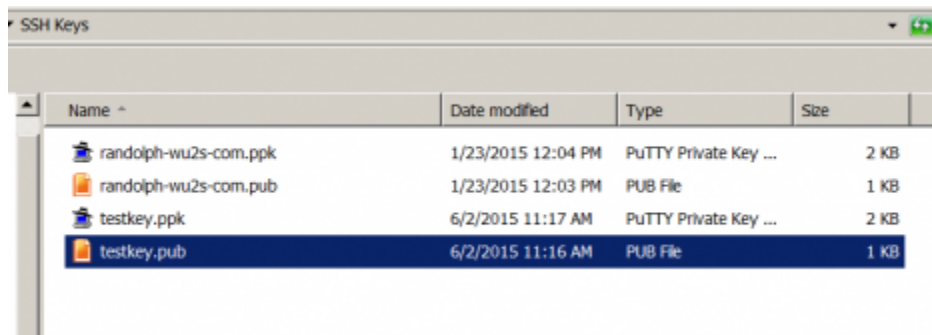
8. Switch back to the PuTTYGen window again and select the “Save Private Key” button. This will let you save the private key just as you did in the previous step with the public key. You are finished generating and saving your SSH keys. Exit PuTTYGen.



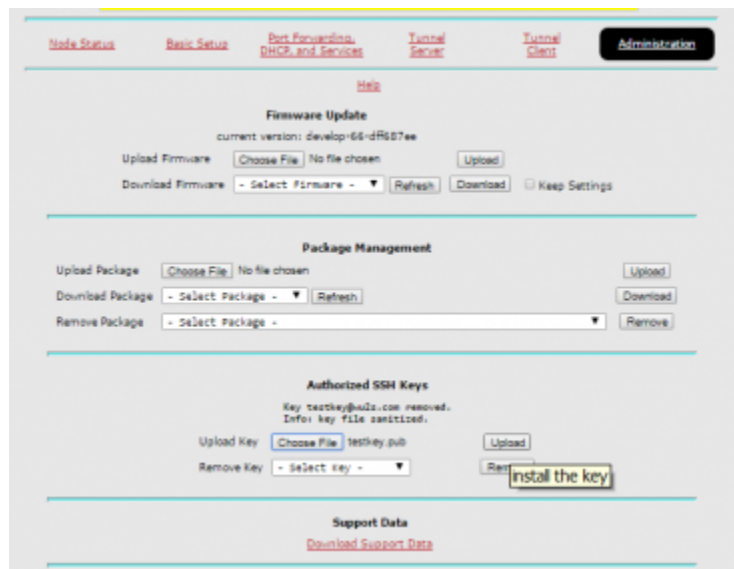
9. In order to use your new SSH key pair, login to your AREDN® node and go to the *Setup -> Administration* screen. At the bottom you will see the Authorized SSH Keys section where you will install the public keys to use on this node.



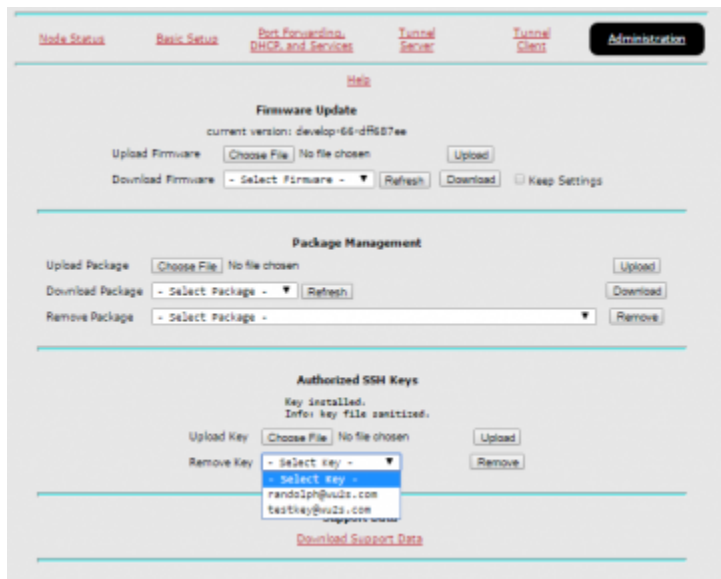
- When you press the Select File button you see a dialog box which enables you to locate the public SSH key that you want to install.



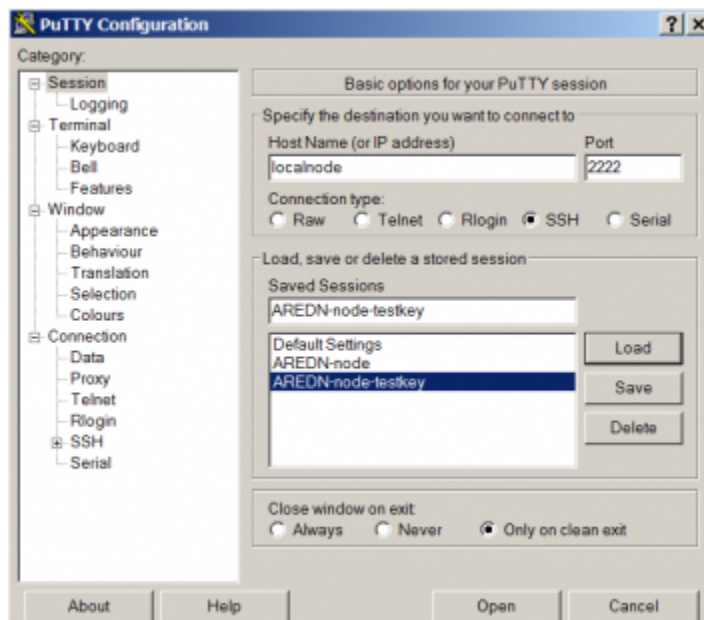
- After choosing the desired public key file. Select the *Upload* button to install the key on the AREDN® node.



12. After installing the new public key, confirm that it is ready for use by looking in the dropdown list at the *Remove Key* section. If your SSH key filename appears, then it is installed properly. DO NOT remove it. In the example below there are two SSH keys currently installed on this node.

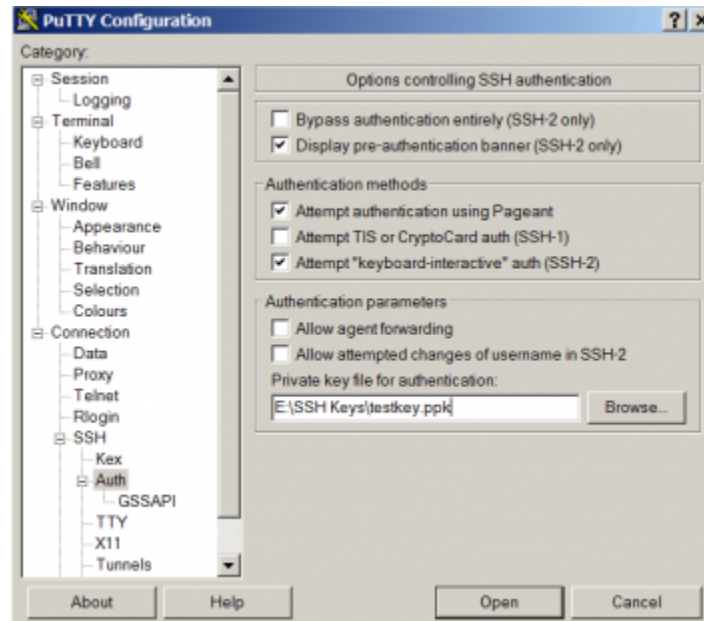


13. To use your SSH keys, open a new PuTTY session. In the Hostname box enter *localnode* and in the Port box enter 2222. It is helpful to save this session definition as something you will remember. Select the Save button.

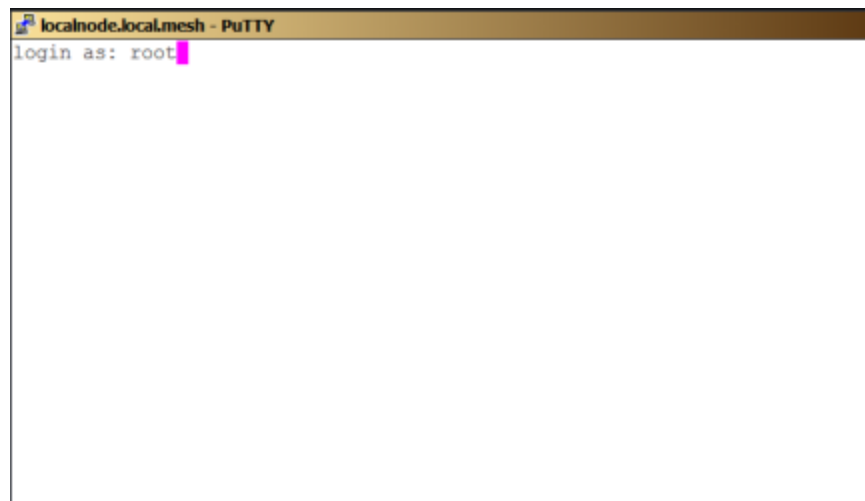


14. Now, using the menu at the left, go to the SSH section and then select the *Auth* item. This shows a number of Options. The only one we need is the very last – the location of the Private key file for authentication. Browse for it and select the correct filename as before. Remember that the PRIVATE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

the session definition again.



15. 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 session box as shown below. Login to the AREDN® node as *root*.



16. If you configured the PuTTY session correctly, it will find your private key file and ask you for the passphrase. If PuTTY cannot find the private key file, it will revert to prompting you for the *root* password that you normally use on the node.



```
localnode.localmesh - PuTTY
login as: root
Authenticating with public key "testkey@wu2s.com"
Passphrase for key "testkey@wu2s.com": 
```

17. 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.

```
localnode.localmesh - PuTTY
login as: root
Authenticating with public key "testkey@wu2s.com"
Passphrase for key "testkey@wu2s.com":

BusyBox v1.28.3 () built-in shell (ash)

      ^ _ | _ \ | _ | _ \ | TM
    / \ | | ) | | | | | | 
   / ^ | | - / | | | | | | 
  / _ | | | \ | | | | | | 
 / \ | \ | \ \ | _ | | | | 
AMATEUR RADIO EMERGENCY DATA NETWORK

-----
* 1 Battery          Connect all devices
* 2 POE injectors    Upgrade firmware to AREDN
* 3 cat5 cables       Setup with your callsign
* 1 UBNT NanoStation Point the Antenna
* 1 IpCam             Welcome to the Mesh!
-----

root@WU2S-CPE5-72-125-44:~# cat sysinfo/model
TP-Link CPE510 v1.0
root@WU2S-CPE5-72-125-44:~#
```


CHAPTER 24

Settings for Radio Mobile

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. The full AREDN® forum post for these settings appears here: [Radio Mobile Settings](#)

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 (μ 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 μ V for the Receive Threshold will approximate the device's receive sensitivity of -94 dB. It is usually best to underestimate the TX and RX Antenna Gain in order to obtain a more realistic model.

When Radio Mobile completes its link analysis, it will display the Fade Margin. For a solid connection a fade margin of 15 dB or greater is needed. Anything above that will only increase the MCS rate. For example, MCS15 requires 19 dB more received signal (94 - 75) and the Ubiquiti Rocket transmit power is 5 dB lower at that same rate, so you will need a total of 24 dB (19 + 5) additional fade margin (39 dB in total) to achieve that data rate. 39 dB is a large Fade Margin and

is not often achieved on a link.

Determining the MCS Rate

If you telnet to your node, the following command will indicate the MCS rate the device is running:

```
cat /sys/kernel/debug/ieee80211/phy0/netdev:wlan0/stations/*/rc_stats
```

Here is an example from an endpoint node pointing to a backbone node over 25 miles away. The *Node Status* screen indicates -73/-95/22 dB SNR.

>>>

type	rate	throughput	ewma	prob	this	prob	retry	this
↪succ/attempt	success	attempts						
HT20/LGI	MCS0	5.6	100.0	100.0	1			
↪ 0 (0)	1	1						
HT20/LGI	MCS1	10.5	100.0	100.0	4			
↪ 0 (0)	4	4						
HT20/LGI	MCS2	14.8	100.0	100.0	5			
↪ 0 (0)	93	93						
HT20/LGI	MCS3	18.6	97.7	100.0	5			
↪ 0 (0)	1380	1416						
HT20/LGI tP MCS4		25.1	99.9	100.0	5			
↪ 0 (0)	31688	33264						
HT20/LGI	MCS5	8.6	25.8	100.0	0			
↪ 0 (0)	175	3495						
HT20/LGI	MCS6	0.0	0.0	0.0	0			
↪ 0 (0)	1	3495						
HT20/LGI	MCS7	0.0	0.0	0.0	0			
↪ 0 (0)	0	3495						
HT20/LGI	MCS8	10.5	100.0	100.0	0			
↪ 0 (0)	1	1						
HT20/LGI	MCS9	18.6	99.9	100.0	5			
↪ 0 (0)	368	380						
HT20/LGI	MCS10	25.1	99.9	100.0	5			
↪ 0 (0)	37921	38776						
HT20/LGI T MCS11		30.3	99.9	100.0	5			
↪ 0 (0)	439091	448760						
HT20/LGI	MCS12	14.1	33.2	100.0	6			
↪ 0 (0)	4482	8447						
HT20/LGI	MCS13	0.0	0.0	0.0	0			
↪ 0 (0)	0	3495						
HT20/LGI	MCS14	0.0	0.0	0.0	0			
↪ 0 (0)	0	3496						
HT20/LGI	MCS15	0.0	0.0	0.0	0			
↪ 0 (0)	0	3495						

The “T” in the 10th character position indicates the current MCS rate, and a “t” indicates the

current fallback rate. In this case the link is running MCS11 at 30.3 Mbps.

CHAPTER 25

Test Network Links with iperf

`iperf` is a network bandwidth testing tool which is available as an AREDN® package for use on mesh nodes. It is a client-server utility, so it must be available on each node that will participate in the network test scenario. The `iperf` client node generates traffic which is sent to the server node. TCP bandwidth is measured and an estimate of the network speeds between that client and server is displayed.

Understand the impact to your network before using `iperf`. During the test period `iperf` 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 `iperf` testing during times when you know that there will be minimal impact to users and routine traffic on your network.

25.1 Installing iperf and IperfSpeed

Two packages should be installed on each AREDN® node in order to facilitate testing between nodes. The `iperf3` package allows the nodes to function either as an `iperf` client or server during the test. The `iperfspeed` package provides a web-based control interface for running network tests between the nodes.

25.2 Using IperfSpeed

After `iperf` and `IperfSpeed` are installed on your nodes, you can select the *IperfSpeed* service on one of the nodes to open its web interface in a new browser tab. From the dropdown lists, select

a node as the iperf server and also one as the iperf client. Click the *Run Test* button to begin the network bandwidth test.

Run a Iperf Speed Test

Server:

kc0euw-nl2

Client:

kc0euw-2-o-portable

RUN TEST

Test Results

```

Starting iperf server
iperf server started
Starting iperf client
Connecting to host kc0euw-nl2, port 5201
[ 5] local 10.136.70.200 port 53126 connected to 10.22.15.88 port 5201
[ ID] Interval            Transfer    Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec   638 KBytes  5.22 Mbits/sec    0   48.1 KBytes
[ 5]  1.00-2.00    sec   472 KBytes  3.87 Mbits/sec    0   53.7 KBytes
[ 5]  2.00-3.00    sec   588 KBytes  4.82 Mbits/sec    0   53.7 KBytes
[ 5]  3.00-4.00    sec   691 KBytes  5.66 Mbits/sec    0   66.5 KBytes
[ 5]  4.00-5.00    sec   564 KBytes  4.62 Mbits/sec    0   66.5 KBytes
[ 5]  5.00-6.00    sec   568 KBytes  4.66 Mbits/sec    0   66.5 KBytes
[ 5]  6.00-7.00    sec   696 KBytes  5.70 Mbits/sec    0   110 KBytes
[ 5]  7.00-8.00    sec   732 KBytes  6.00 Mbits/sec    0   110 KBytes
[ 5]  8.00-9.00    sec   602 KBytes  4.94 Mbits/sec    0   110 KBytes
[ 5]  9.00-10.00   sec   833 KBytes  6.82 Mbits/sec    0   110 KBytes
-----
[ ID] Interval            Transfer    Bitrate      Retr
[ 5]  0.00-10.00   sec   6.24 MBytes  5.23 Mbits/sec    0          sender
[ 5]  0.00-10.08   sec   6.16 MBytes  5.13 Mbits/sec              receiver

```

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 of the results from the perspective of the sender (client) and the receiver (server). 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.

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

CHAPTER 26

Frequencies and Channels

The frequencies and channels that are available for AREDN® networking are shown in the charts below.

2.4 GHz

2.4 GHz	Channel	-2	-1	0*	1	2	3	4	5	6
	Status	Ham Band			Shared Ham and ISM/WiFi Band					
	Freq	2.397	2.402	2.407	2.412	2.417	2.422	2.427	2.432	2.437

*Not available for use

3.4 GHz

3.4 GHz	Channel	76	77	78	79	80	81	82	83	84	85	86	87
	Status	Ham Band											
	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

88	89	90	91	92	93	94	95	96	97	98	99
3.440	3.445	3.450	3.455	3.460	3.465	3.470	3.475	3.480	3.485	3.490	3.495

Refer to your local band plan for coordination

5.8 GHz

5.8 GHz	Channel	133	134	135	136	137	138	139	140	141	142	143	144	145
	Status	Ham Band shared with U-NII-2C/wifi/unlicensed												
	Freq	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
		146	147	148	149	150	151	152	153	154	155	156	157	158
		Ham Band shared with U-NII-3/wifi/unlicensed												
		5.730	5.735	5.740	5.745	5.750	5.755	5.760	5.765	5.770	5.775	5.780	5.785	5.790
		159	160	161	162	163	164	165	166	167	168	169	170	171
		Ham Band shared with U-NII-3/wifi/unlicensed												
		5.795	5.800	5.805	5.810	5.815	5.820	5.825	5.830	5.835	5.840	5.845	5.850	5.855
		172	173	174	175	176	177	178	179	180	181	182	183	184
		Ham Band												
		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

Refer to your local band plan for coordination; ★ 5825 to 5850 Shared under Part 15.247 with a limited number of WISP operators and may be encountered at tower sites

CHAPTER 27

Additional Information

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

- [AREDN homepage](#)
- [AREDN forums](#)

27.1 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 a username, email, and password. We suggest using your callsign as the username.
3. On the GitHub website, click the `Sign In` button and enter your username or email followed by your GitHub password.
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 identical to the workflow for contributing code which is described in the file titled **How to Use GitHub for AREDN**, a copy of which you already have in your new local repository. Refer to that document for additional information about contributing AREDN® documentation.

The only difference is the repository name of `aredn/documentation` and the main branch name of `master`. 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 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