# AREDN Documentation

*Release 3.19.3.0*

**AREDN**

**Jul 05, 2020**

# Guía de inicio


**Version** 3.20.3.0

Esta documentación está compuesta de diferentes secciones, las cuales se muestran en el panel de navegación.

- La **Guía de inicio** recorre el proceso de configuración de un nodo radio de AREDN® como parte de una red mallada.
- La **Guía de diseño de red** proporciona información general y consejos para planificar y desplegar una red mallada de forma robusta.
- La **Guía de aplicaciones y servicios** analiza los tipos de programas o servicios que pueden ser utilizados a través de la red mallada.
- Las **Guías de procedimiento ('How-to')** contienen consejos y técnicas referentes a conceptos variados.
- Por último, en el **Apéndice** se incluye información suplementaria.

Si necesitas localizar algún apartado concreto dentro de la documentación, puedes buscar por palabras clave utilizando el campo *Search docs*. Este mecanismo te mostrará el listado de elementos que se ajustan a los criterios de búsqueda.

Por otro lado, si necesitas revisar la documentación específica de una versión de AREDN®, pincha en el botón etiquetado como *Read the Docs*, situado al final del panel de navegación. Se te mostrarán diferentes opciones, entre las que se encuentran consultar versiones anteriores, o descargar la documentación para consultar offline en cualquiera de los formatos soportados (*PDF*, *Epub*, *HTML*, . . . )

---

**Note:** AREDN® es una marca registrada de *Amateur Radio Emergency Data Network, Inc.*

---

# CHAPTER 1

# Información general de AREDN®

AREDN® es un acrónimo que significa *Amateur Radio Emergency Data Network* (cuya traducción aproximada al español sería *Red de Datos de Emergencia Radio Amateur*). Proporciona los mecanismos necesarios para que operadores de radio amateur establezcan una red de datos de alta velocidad `ad-hoc`, que podrá ser utilizada tanto por comunicaciones de emergencia como por comunicaciones orientadas a servicio.

Durante muchos años, los operadores de radio amateur, así como sus agencias relacionadas, han confiado sus comunicaciones de emergencia a las transmisiones de voz. Un escenario típico de intercambio de mensajes involucraría la transmisión de la información a un operador radio, que la anotaría en un formulario estándar ISC-213. Este mensaje sería retransmitido de nuevo, involucrando a un nuevo operador en recepción y su correspondiente formulario ISC-213. El destinatario final recibiría el formulario en mano, para su lectura y firmado. Cualquier acuse de recibo o respuesta se manejaría de este mismo modo, pero en sentido opuesto.

Este escenario ha demostrado con creces que puede funcionar bien y, de hecho, todavía continua sostentando mucho tráfico de emergencia. Sin embargo, hoy en día la transmisión digital se ha convertido en un mecanismo mucho más común frente a estos métodos tradicionales. Y afecta tanto la parte del formulario (cuyo modelo físico ISC-213 está siendo relegado a un segundo plano en pro de formulario electrónico Winlink), como a la parte de la tecnología de transmisión (AX.25, HF Pactor, Fldigi, . . . )

**Nuestra misión**

El objetivo principal del proyecto AREDN® es el de posibilitar a operadores de radio amateur con licencia el despliege rápido y sencillo de redes de datos de alta velocidad, cuándo y dónde

sean necesarias.

Dentro del escenario tecnológico social actual, la gente se ha acostumbrado a manejar sus necesidades de comunicación de múltiples maneras. En primer lugar se situan el intercambio de mensajes cortos y la comunicación *teclado a teclado*, seguidos por la videollamada sobre VoIP (Voz sobre IP) y otras tecnologías de streaming.

La comunidad radio amateur es capaz de alcanzar los requisitos necesarios de banda ancha mediante el uso del espectro de frecuencia recogido en la FFC Part 97, consiguiendo establecer una red de datos tolerante a fallos y autoregenerativa. Este entorno, de algún modo, ha sido descrito como la versión radio amateur de Internet. Y, aunque no está concebida para facilitar conexión a Internet, sí que provee acceso a aplicaciones típicas de Internet/intranet para su uso durante eventos comunitarios y/o situaciones de emergencia.

Una red AREDN® es capaz de proporcionar el mecanismo de transporte para cualquier aplicación de comunicación, dentro de un flujo normal de negocio e interacción social (email, chat, servicio telefónico, intercambio de ficheros, videoconferencia, ...). Dependiendo de las características particulares del despliegue, la red podría operar a velocidad cercanas a Internet con distancias de decenas de kilómetros entre nodos.

El objetivo principal del proyecto AREDN® es el de posibilitar a operadores de radio amateur con licencia el despliege rápido y sencillo de redes de datos de alta velocidad, buscando prestar servicio tanto a la comunidad general, como a la afición en particular. Esto es especialmente importante cuando los servicios esenciales tradicionales dejan de estar disponibles (electricidad, lineas telefónicas, o Internet). En esos casos, una *Red de Datos de Emergencia Radio Amateur* puede actuar como salvavidas para una comunidad impactada por un desastre.

# CHAPTER 2

# Seleccionando el Hardware de Radio

La comunidad de radioaficionados ha reconocido los beneficios de utilizar radios comerciales de bajo costo: abbr: *WISP (Wireless Internet Service Provider)* para crear la red AREDN | trade |. Cada uno de estos dispositivos viene con su firmware del fabricante preinstalado, pero siguiendo unos sencillos pasos, este firmware se puede reemplazar con una imagen de firmware hecha por el equipo de AREDN | trade |. Se han adaptado y mejorado varias funciones de software de código abierto para crear esta versión de firmware, basándose en *OpenWRT (enrutador inalámbrico abierto) <https://en.wikipedia.org/wiki/OpenWRT> _* y *OLSR (protocolo de enrutamiento de estado de enlace optimizado) <https://en.wikipedia.org/wiki/ Optimized_Link_State_Routing_Protocol> '_. El equipo de AREDN | trade | crea imágenes de firmware específicas adaptadas a cada modelo radio, y mantienen una lista de los dispositivos compatibles en: 'Matriz de plataformas soportadas <https://www.arednmesh.org/content/supported-platform-matrix/> _.*

Para seleccionar tu dispositivo para AREDN® deberías de tener en consideración lo siguiente:r decision.

- Las radios deben de comprarse teniendo en cuenta la banda de frecuencias en la que van a operar * Actualmente AREDN® soporta dispositivos en las bandas de: 900 MHz, 2.4 GHz, 3.4 GHz y 5.8 GHz.
- Varios dispositivos vienen con un sistema incorporado de MIMO (Multiple Input-Multiple Output), esto ayuda a reducir los problemas de multipath que son típicos en ciudad.
- La mayoria de radios pueden comprarse de forma separada a la antena, así que es posible tener más de una opción de antena para cada radio. De esta manera podremos adaptar el comportamiento de nuestros dispositivos AREDN® a cada ocación.
- El coste de los dispositivos puede ser desde menos de 50€ a varios cientos de euros por un

nodo completo. Hay una opción para cada bolsillo. El mercado de segunda mano para este tipo de radios es muy activo.

- Algunos dispositivos antiguos tienen una cantidad limitada de recursos (ya sea en forma de memoria de programa o ram) que podrían traducirse en el abandono del soporte por parte del equipo de AREDN®. Ten esto en cuenta a la hora de elegir un dispositivo en la matriz de plataformas soportadas.
- Comprueba que la potencia de radio del dispositivo sea suficiente para el enlace que quieres hacer, teniendo en cuenta la normativa local.

Uno de las mejores fuentes de información detallando las caracteristicas técnicas de cada dispositivo es su fabricante. Actualmente AREDN® soporta más de cincuenta modelos, incluyendo fabricantes como: GL-iNET, Mikrotik, TP-LINK, o Ubiquiti Networks.

Si estás empezando a interesarte por AREDN® puedes empezar fácilmente con uno de los dispositivos de menor coste, habitualmente con la antena integrada y POE (Power over Ethernet). Ya llegará eel momento de expandir tu red con equipo más sofisticado.

---

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

---

# CHAPTER 3

# Descargando el Firmware AREDN®

Una vez seleccionado y obtenido un dispositivo, el siguiente paso es elegir la imagen de firmware AREDN® que coincida con ese dispositivo específico. La página de descarga AREDN muestra las versiones de firmware más actuales para todos los dispositivos soportados.

Localice modelo/versión de su dispositivo en la columna de la izquierda. Muchos fabricantes ponen la versión del hardware en la etiqueta de su producto. En otros casos, puede ser necesario arrancar el dispositivo con su firmware pre-instalado y navegar a la página de información del sistema para determinar la versión del hardware.

Hay dos tipos de imagen de firmware: una para la primera vez que se reemplaza el firmware del fabricante, y la otra para actualizaciones de los nodos que ya están funcionando con un firmware AREDN®.

- Si está cargando el firmware AREDN® en un dispositivo por primera vez, debe descargar el firmware *factory* desde la columna central. Para los dispositivos Mikrotik, debe descargar, además, la imagen *sysupgrade* de la columna de la derecha.
- Si ya está ejecutando el firmware AREDN® en el nodo, entonces seleccione el firmware *sysupgrade* de la columna de la derecha y después utilizará la interfaz web AREDN® para realizar la actualización del firmware.

Una vez que haya seleccionado la imagen de firmware correcta para su dispositivo, haga clic en el enlace para descargar el archivo a su ordenador local. Tome nota de la ubicación de descarga en su computadora, ya que necesitará usar esa imagen para instalar el firmware AREDN® en su dispositivo.

# CHAPTER 4

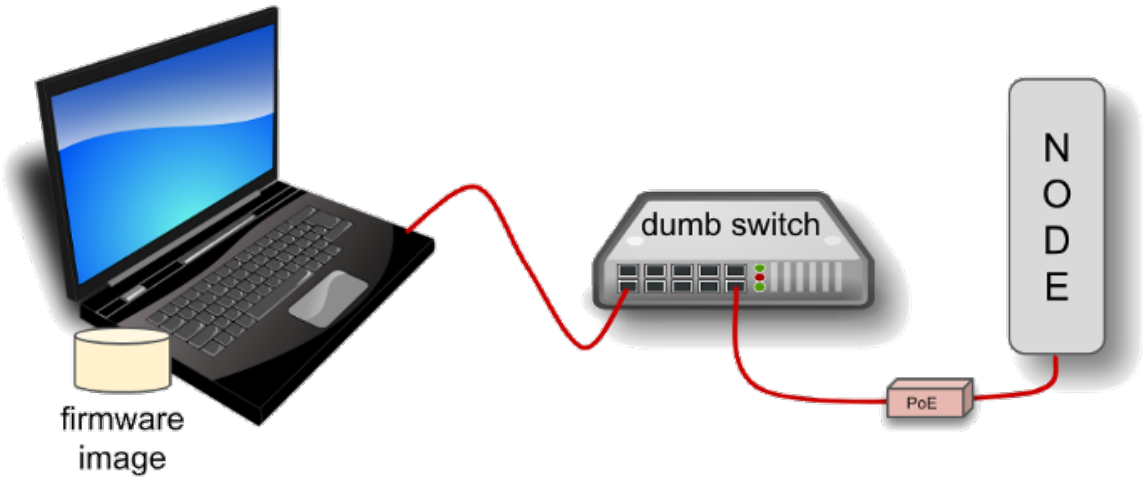
# Instalación del Firmware AREDN®

Los pasos para instalar el firmware en un dispositivo están documentados en el sitio web de AREDN® en la sección Current Software. En el menú **Software**, seleccione **Download** para acceder a la página *Current Software*.

Hay dos casos para instalar el firmware AREDN®:

1. Si ya tiene una versión de firmware de AREDN® ejecutándose en su dispositivo, puede usar la interfaz web de su computadora para navegar a **Setup > Administration > Firmware Update** para instalar su nuevo firmware. Este proceso se explicará con más detalle en la sección **Configuración Avanzada** de esta guía. Para obtener información adicional vea también *Consejos de Actualizacióón de Firmware* en la sección **Guías Prácticas**.

2. Si está instalando el firmware AREDN® en un dispositivo por primera vez, cada plataforma de hardware puede requerir un procedimiento único.




El diagrama anterior muestra que su ordenador con la imagen de firmware descargada debe estar conectado al nodo mediante cable Ethernet para instalar la imagen AREDN®. Es Útil conectar ordenador y nodo a través de un simple switch Ethernet para que éste pueda mantener el enlace con el ordenador mientras se reinicia el nodo.

Nodos con diferente hardware requerirán métodos diferentes para instalar el firmware AREDN®. Para los dispositivos Ubiquiti, el cliente TFTP de su computadora se conectará al servidor TFTP del nodo para cargar la imagen del firmware. Para los dispositivos TP-LINK y GL-iNET, el navegador web de su ordenador se conectará al servidor web del nodo para cargar la imagen del firmware. Para los dispositivos Mikrotik, su ordenador ejecutará un servidor de arranque remoto y el cliente de arranque remoto del nodo cargará su imagen de arranque desde su computadora. Consulte los procedimientos específicos a continuación para el hardware de su nodo.

## 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 another standalone TFTP client of your choice.

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

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

1. Set your computer's Ethernet network adapter to a static IP address that is a member of the correct subnet for your device. Check the documentation for your specific hardware

to determine the correct network number. As in the example below, most Ubiquiti devices have a default IP address of 192.168.1.20, so you can give your computer a static IP on the 192.168.1.x network with a netmask of 255.255.255.0. For example, set your Ethernet adapter to a static IP address of 192.168.1.100.

You can choose any number for the fourth octet, as long as it is not the same as the IP address of the node. Of course you must also avoid using 192.168.1.0 and 192.168.1.255, which are reserved addresses that identify the network itself and the broadcast address for that network. Other devices may have different default IP addresses or subnets, so select a static IP for your computer which puts it on the same subnet but does not conflict with the default IP of the device.

2. Connect an Ethernet cable from your computer to the dumb switch, and another cable from the LAN port of the PoE adapter to the switch.
3. Put the Ubiquiti device into TFTP mode by holding the reset button while plugging your node's Ethernet cable into the POE port on the PoE adapter.
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 a file transfer command to send the AREDN firmware to your device. Target the default IP address of your Ubiquiti node, such as 192.168.1.20 or 192.168.1.1 for AirRouters. The following is one example of TFTP commands that transfer the firmware image to a node:

```
>>>
[Linux/Mac]
> tftp 192.168.1.20
> bin                  [Transfer in "binary" mode]
> trace on             [Show the transfer in progress]
> put <full path to the firmware file>
  [For example, put /temp/aredn-3.19.3.0-ubnt-nano-m-xw-
 ↪factory.bin]
-----------------------------------
[Windows with command on a single line]
> tftp -i 192.168.1.20 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. You may need to unplug/reconnect the Ethernet cable from your computer to force it to get a new IP address from the node.

8. After the node reboots, open a web browser and enter the following URL: `http://localnode.local.mesh` Some computers may have DNS search paths configured that require you to use the fully qualified domain name (FQDN) to resolve *localnode* to the mesh node's IP address.
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 that is a member of the correct subnet for your device. Check the documentation for your specific hardware to determine the correct network number. As in the example below, most TP-LINK devices use the 192.168.0.x subnet by default, so you can give your computer a static IP such as 192.168.0.100 with a netmask of 255.255.255.0.

   You can choose any number for the fourth octet, as long as it is not the same as the IP address of the node and is not within the range of DHCP addresses you will be providing in step 2 below. Of course you must also avoid using 192.168.0.0 and 192.168.0.255, which are reserved addresses that identify the network itself and the broadcast address for that network. Other devices may have different default IP addresses or subnets, so select a static IP for your computer which puts it on the same subnet.

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. Use a `dhcp-range` of IP addresses that are also on the same subnet as the computer: for example 192.168.0.110,192.168.0.120 as shown below.
3. Become `root` and open a terminal window to execute the following dnsmasq command:

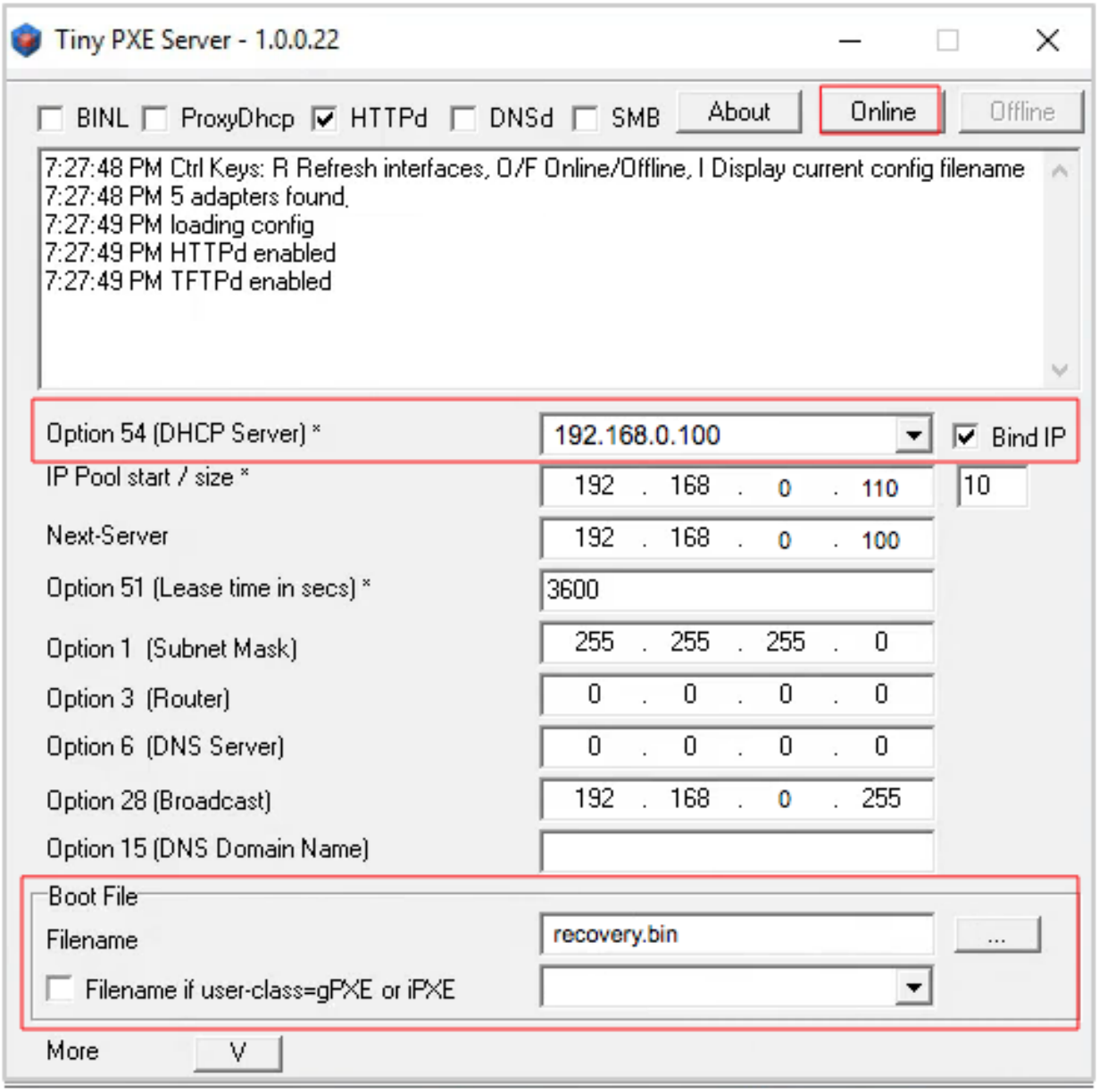
```
>>>
(root)# dnsmasq -i eth0 -u joe --log-dhcp --bootp-dynamic \
      --dhcp-range=192.168.0.110,192.168.0.120 -d -p0 -K \
      --dhcp-boot=recovery.bin --enable-tftp --tftp-root=/tftp/
```

4. With the 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 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` Some computers may have DNS search paths configured that require you to use the fully qualified domain name (FQDN) to resolve *localnode* to the mesh node's IP address.
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 correct mikrotik-vmlinux-initramfs file with the **elf** extension, and then use the in-memory-only AREDN® Administration UI to complete the installation of the appropriate mikrotik-rb file with the **bin** extension.

**Preparation**

1. Download the appropriate Mikrotik **elf** and **bin** files. Rename the *elf* file to `rb.elf` and keep the *bin* file available for later.
2. Set your computer's Ethernet network adapter to a static IP address that is a member of the correct subnet for your device. Check the documentation for your specific hardware to determine the correct network number. As in the example below, most Mikrotik devices use the 192.168.1.x subnet by default, so you can give your computer a static IP such as 192.168.1.100 with a netmask of 255.255.255.0.

   You can choose any number for the fourth octet, as long as it is not the same as the IP address of the node and is not within the range of DHCP addresses you will be providing in step 2 below. Of course you must also avoid using 192.168.1.0 and 192.168.1.255, which are reserved addresses that identify the network itself and the broadcast address for that network. Other devices may use different default subnets, such as QRT units which use 192.168.88.x. Select a static IP for your computer which puts it on the same subnet as your device.
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 192.168.1.100 above. You will use this interface name in the command below as the name after `-i` and you must substitute your login user name after `-u` below. Use a

dhcp-range of IP addresses that are also on the same subnet as the computer: for example 192.168.1.110,192.168.1.120 as shown below.

3. Become root and open a terminal window to execute the following dnsmasq command:

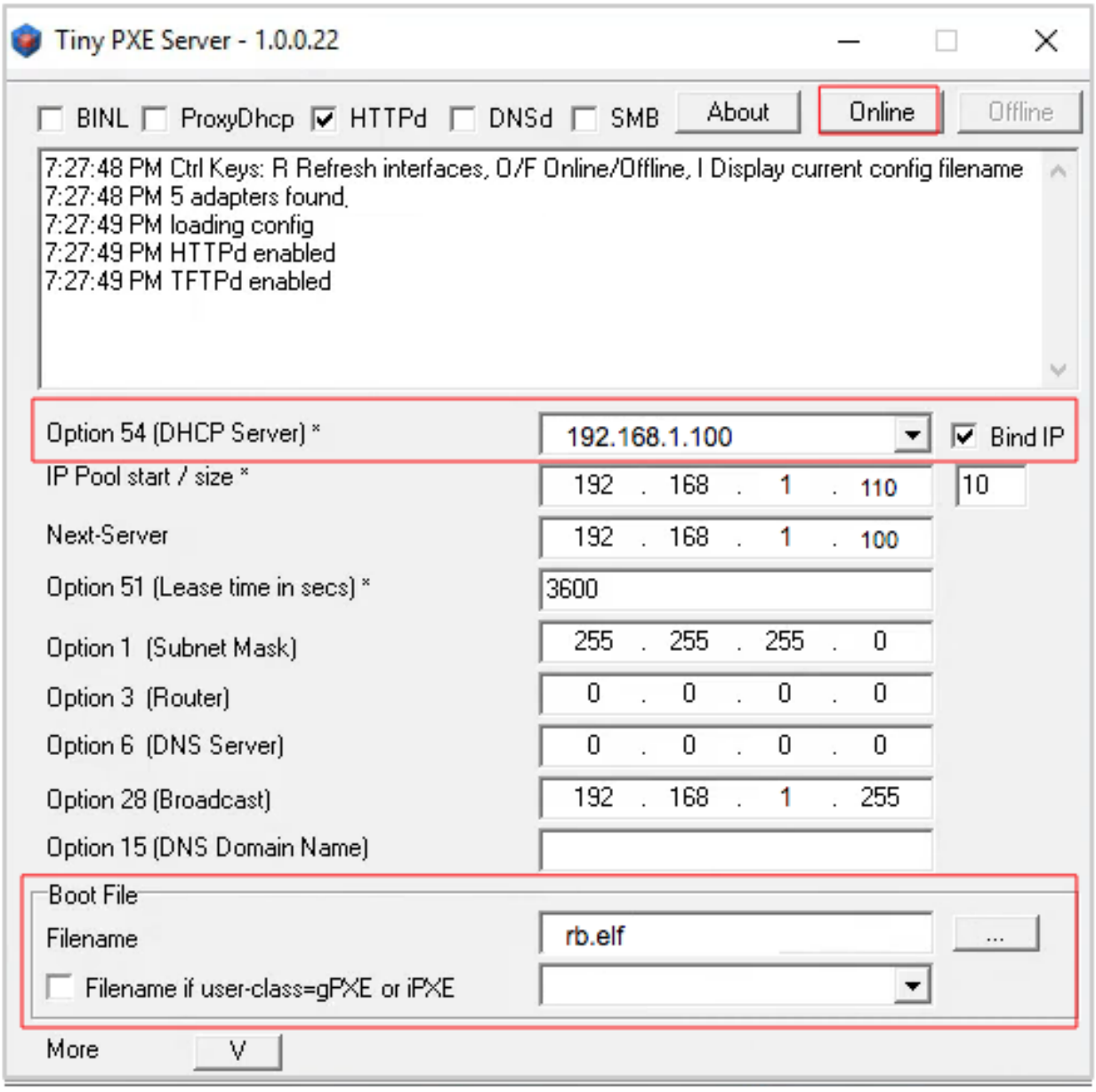
```
>>>
(root)# dnsmasq -i eth0 -u joe --log-dhcp --bootp-dynamic \
      --dhcp-range=192.168.1.110,192.168.1.120 -d -p0 -K \
      --dhcp-boot=rb.elf --enable-tftp --tftp-root=/tftp/
```

4. With the 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 ten 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 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. For the hAP ac lite, pull the Ethernet cable from the WAN port (1) on the Mikrotik and insert it into one of the LAN ports (2,3,4). 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. You may need to disconnect and reconnect your computer's network cable to ensure that your IP address has been reset. Also, you may need to clear your web browser's cache in order to remove cached pages remaining from your node's previous firmware version.

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

   As an alternative to using the node's web interface, if your node has plenty of free memory you can copy the **bin** file to the node and run a command line program to install the image. This will allow you to see any error messages that are not displayed when using the web interface upgrade procedure. Execute the following commands from your computer:

   ```
   >>>
   my-computer:$ scp -P 2222 aredn-firmware-filename.bin root@192.168.
   ↪1.1:/tmp
   my-computer:$ ssh -p 2222 root@192.168.1.1
   ~~~~~~~ after logging into the node with ssh ~~~~~~~
   node:# sysupgrade -n /tmp/aredn-firmware-filename.bin
   ```

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.

## 4.4 GL-iNET First Install Process

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

After the GL-iNET device has been booted and configured, navigate to the *Upgrade* section and click *Local Upgrade* to select the AREDN® "sysupgrade.bin" file you downloaded for your device. Be sure to uncheck/deselect the "Keep Settings" checkbox, since GL-iNET settings are incompatible with AREDN. After the device has rebooted to the AREDN® image, you should be able to navigate to `http://192.168.1.1` for the firstboot or NOCALL page to appear.

If for some reason your GL-iNET device gets into an unusable state, you should be able to recover using the process documented here: GL-iNET debrick procedure

## 4.5 Troubleshooting Tips

One common issue can occur when installing firmware using a web browser interface. The browser cache stores data for the URLs that have been visited, but IP addresses and other parameters often change during the install process. It is possible for the cache to contain information that doesn't match the latest settings for the URL, so the browser may block the connection setup and display an ERR_CONNECTION_RESET message. Clearing the web browser's cache will allow the latest URL settings to be registered so you can continue with the install process.

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

- Refresh or Reload the URL for your node.
- Clear your browser cache and delete cookies.
- Close your browser and restart a new session.
- Use a different web browser program or a *Safe Mode / Incognito* browser window.
- Unplug and reconnect the Ethernet cable from your computer to ensure that your machine has received a new DHCP IP address on the same subnet as the node's updated IP.

If for some reason the node's web interface does not work, you may be able to use a command line program to install the firmware image. You must first copy the firmware *bin* file to the node, then log into the node and use the *sysupgrade* program to install the image as illustrated below.

```
>>>
my-computer:$ scp -P 2222 aredn-firmware-filename.bin root@192.168.1.
↪1:/tmp
my-computer:$ ssh -p 2222 root@192.168.1.1
~~~~~~~ after logging into the node as root (hsmm) ~~~~~~~
node:# sysupgrade -n /tmp/aredn-firmware-filename.bin
```

Additional questions and troubleshooting assistance can usually be obtained by creating a post on the AREDN® online forum, which has an active community of helpful and experienced operators.

## 4.6 Post-Install Steps

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://`

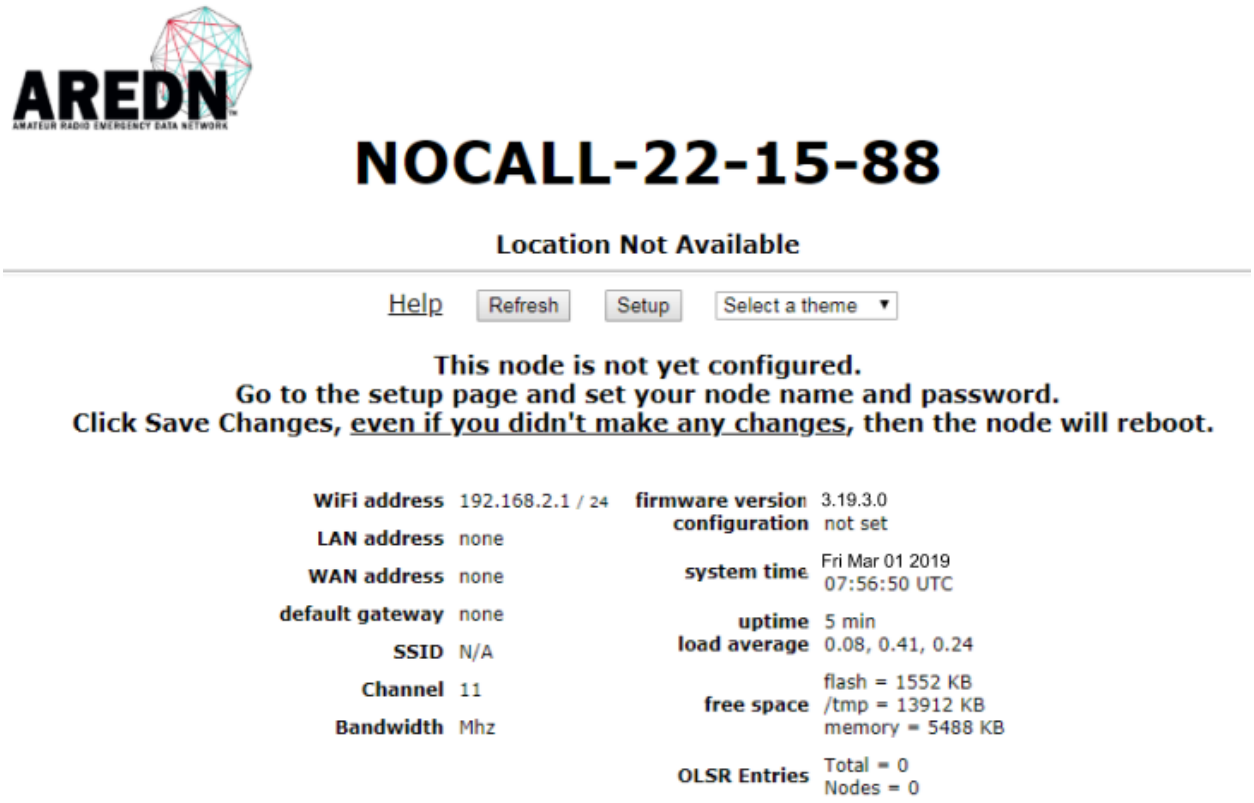
`localnode.local.mesh` Some computers may have DNS search paths configured that require you to use the fully qualified domain name (FQDN) to resolve *localnode* to the mesh node's IP address. Each node will serve its web interface on both port 80 and 8080.

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. You may also need to clear your web browser's cache in order to remove cached pages remaining from your node's previous firmware version.

# CHAPTER 5

# Basic Radio Setup

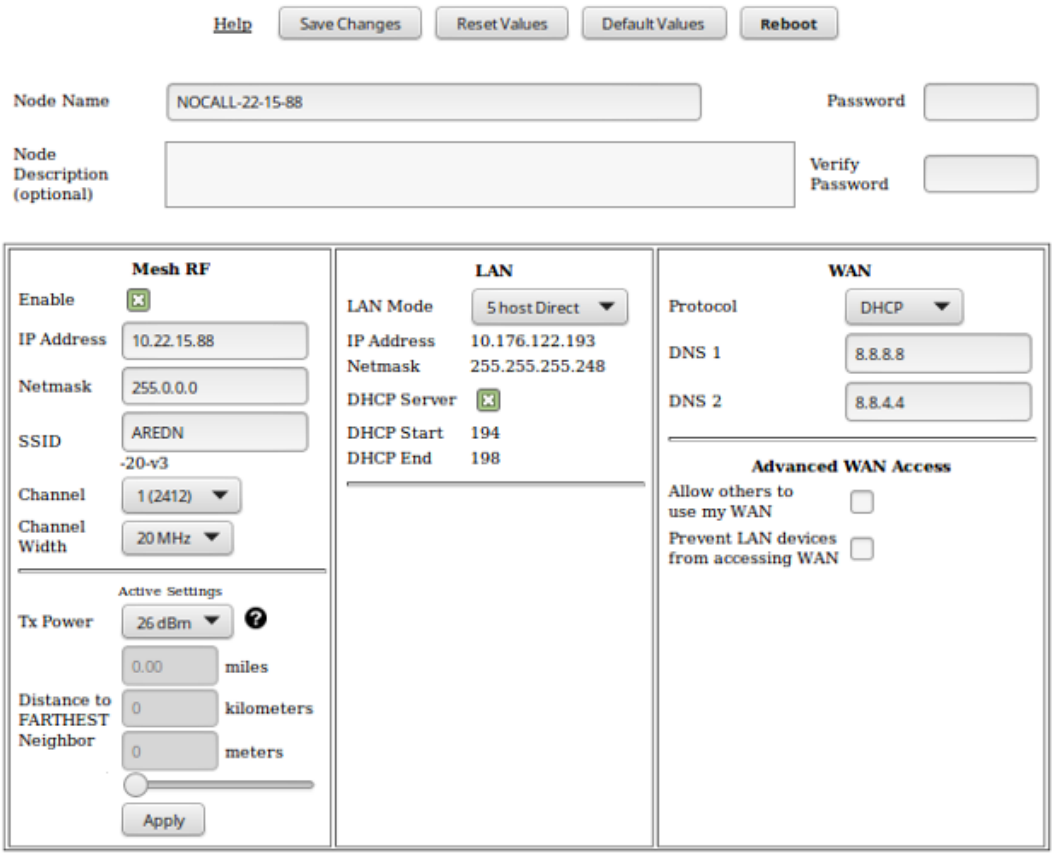
After you have installed the AREDN® firmware, rebooted the device, and connected your computer to the LAN port on the POE you can navigate to the following URL: `http://localnode`. The initial status page will be displayed, instructing you to configure your node by clicking the **Setup** button.




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.




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 beaconed automatically by the node every five minutes, so the node name must contain your callsign. Recommended names follow the (callsign)-(label) format, such as AD5BC-MOBILE or AD5BC-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.

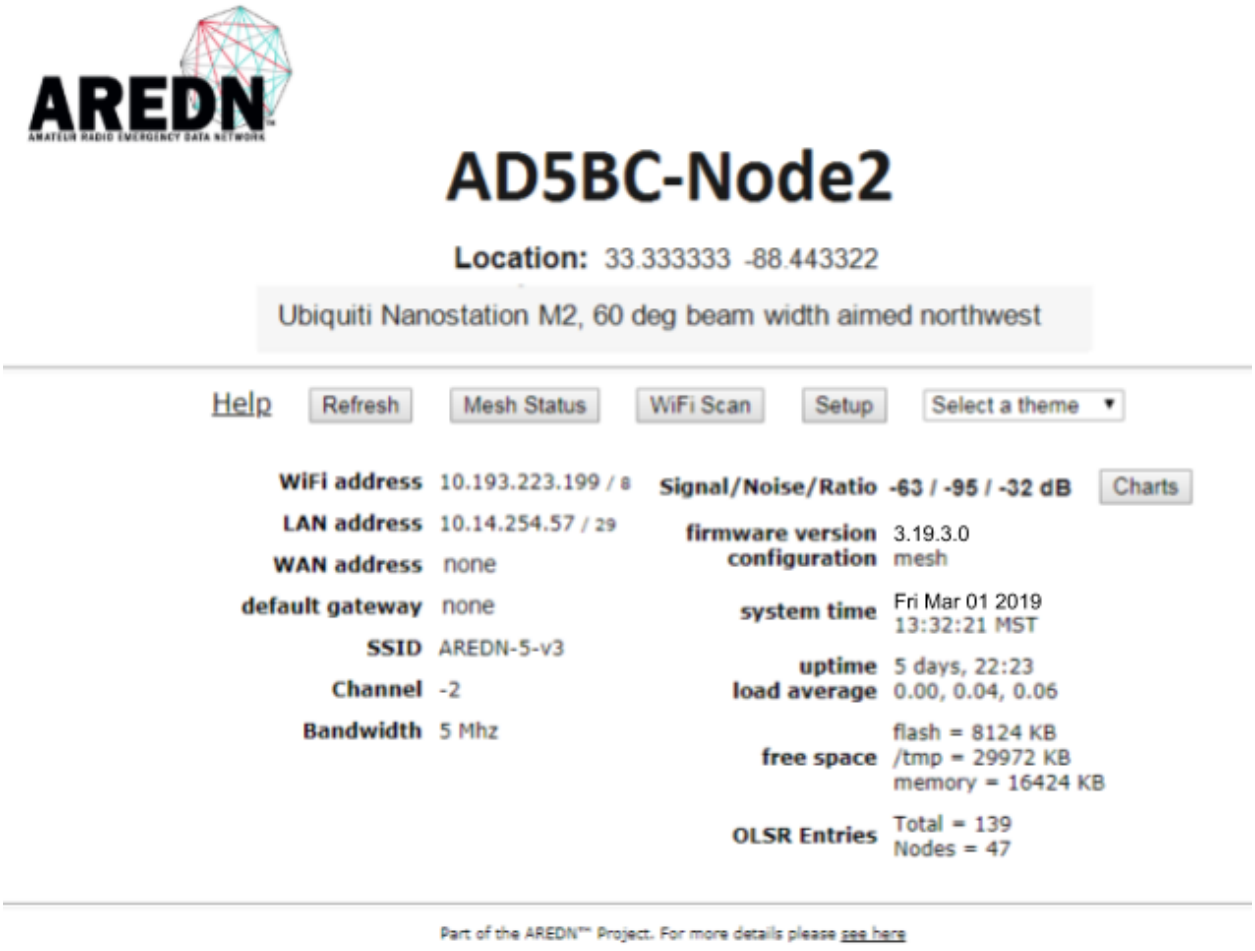
**Optional Settings** In this section you can enter your node's latitude and longitude, as well as the grid square designator. Click the **Apply Location Settings** button after entering this information. You may also change the timezone for your node's system time.

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

# Estado del Nodo

Una vez se ha completado la configuración inicial de tu nodo de AREDN®, puedes conectar tu ordenador al puerto de LAN del inyector POE y navegar a `http://localnode`. Serás redirigido a la página de **Estado de Nodo** tal y como se muestra en la web inferior.




Debajo de la barra con el nombre del nodo, hay varias secciones:

**Ayuda** Abre una nueva ventana para mostrar la ayuda.

**Refresh** Actualiza la información mostrada con datos actuales.

**Mesh Status** Abre la página de **Estado de la Red**, mostrando los nodos cercanos y remotos. Además, muestra los servicios expuestos en esos nodos.

**Escaneo WiFi**

Muestra la lista de otras señales 802.11 (WiFi) que se reciben desde tu nodo. Estas señales incuyen Puntos de Acceso, el WiFi del vecino, y otras redes. Esta función solo muestra dispositivos con la misma configuración de canal (frecuencia y ancho de canal) que la de nuestro nodo. Si estamos instalando en una nueva hubicación, es buena práctca escanear con ancho de canal de 5, 10, and 20MHz en busca de posibles interferencias. Cuando hay más de una red accesible (con diferente SSID o canal), los ID de cada nodo de dicha ned se mostrarán de forma resumida. Además, hay un modo automático de escaneo, pero hacerlo de forma continuada no está recomendado; especialmente si nuestro nodo tiene tráfico.

El escaneo de AREDN es pasivo, solo busca los paquetes que anuncian redes cercanas en los canales. Esto implica que puede perderse alguno. WiFiScan no hace escaneos activos, por lo que no se corre el riesgo de interferir con estaciones de Radar ni DFS cercanos. Varios intentos de escaneo pueden ser necesarios para encontrar todos los dispositivos al alcance.

**Configuración** Abre la página de configuración de tu nodo. Necesitarás usuario y contraseña para acceder a esta página. El usuario siempre es `root`, y la contraseña se configuró la primera vez que se utilizó el nodo. Si el nodo no ha sido configurado nunca, ca contraseña por defecto es `hsmm`.

**Selección de Tema** El firmware AREDN |tradeltiene múltiples temas incluídos por defecto. El standard es `aredn`, pero puedes probar otros que se ajusten mejor a tus preferencias.

## 6.1 Resumen de los ajustes del Nodo

El área bajo los controles muestra dos cosas: En el lado izquierdo contiene los detalles de configuración de este nodo. Concretamente, las direcciones ip de cada interfaz, SSID, frecuencia y ancho de canal.

La columna derecha contiene la intensidad de la sieñal recibida y otros valores de este nodo. La **Relación Señál a Ruido** muestra la señal del nodo vecino más fuerte en DBM (decibels relative to one milliwatt), y estará disponible solo cuando el nodo esté interconectado por RF (Radio Frequency) a la red AREDN | trade |.

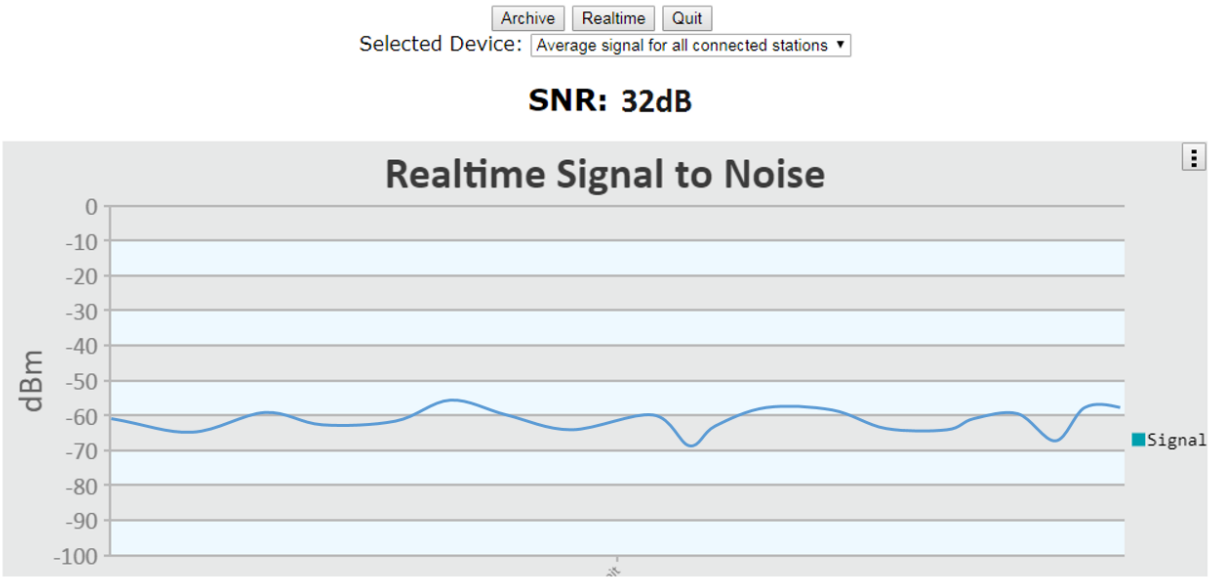
Bajo la lectura de señal, están los datos del sistema tales como: La versión del Firmware, el tipo de red, la hora del nodo, y el tiempo que lleva sin ser reiniciado. Los nodos no tienen ni batería ni un reloj interno. Esto provoca que la hora se pierda cada vez que se reinicia. Si hay salida a internet, tomará la hora usando el protocolo NTP (Network Time Protocol)

La **Carga Media** es la media del número de procesos que tienen lugar en el nodo en los últimos 1, 5 y 15 minutos. El **Espacio Libre** muestra cuanta memoria nos queda en el dispositivo. Esta es la memoria flash del dispositivo, que tiene permanencia entre reinicios. Aquí quedan guardadas las configuraciones y los paquetes de software. La **Memoria** nos indica la cantidad de memoria RAM (Random Access Memory) disponible en el nodo.

Las **Entradas :abbr:‘OLSR (Optimized Link State Routing protocol)‘** Muestran el total de nodos en la tabla de rutas actual. Estos son los nodos de la red AREDN con los que tenemos conectividad.

## 6.2 Signal Charts

There is a **Charts** button next to the node’s **Signal Stength** 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.

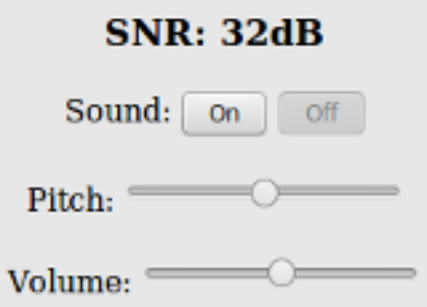
Below these controls you can choose to view the signal strength statistics for individual nodes that are directly connected to your node. Choose the neighbor node from the **Selected Device** dropdown list. Changing the selected device will automatically reload the chart to show that node's information.

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.

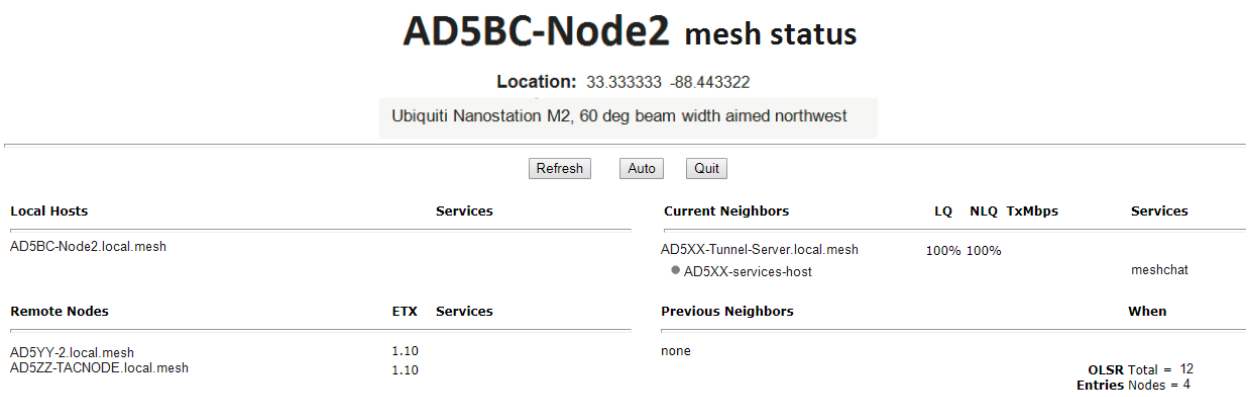



On the left of the Realtime Graph there is an **SNR Sound** control. Clicking the *On* button will cause your computer to emit a tone that corresponds to the relative SNR level, with higher pitch tones indicating better SNR. This feature was added in order to provide an audio queue to operators in the 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.

CHAPTER 7

# Mesh Status Display

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




Below the node name bar there are several controls.

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

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

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

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

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

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

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

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

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

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

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

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

# 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.

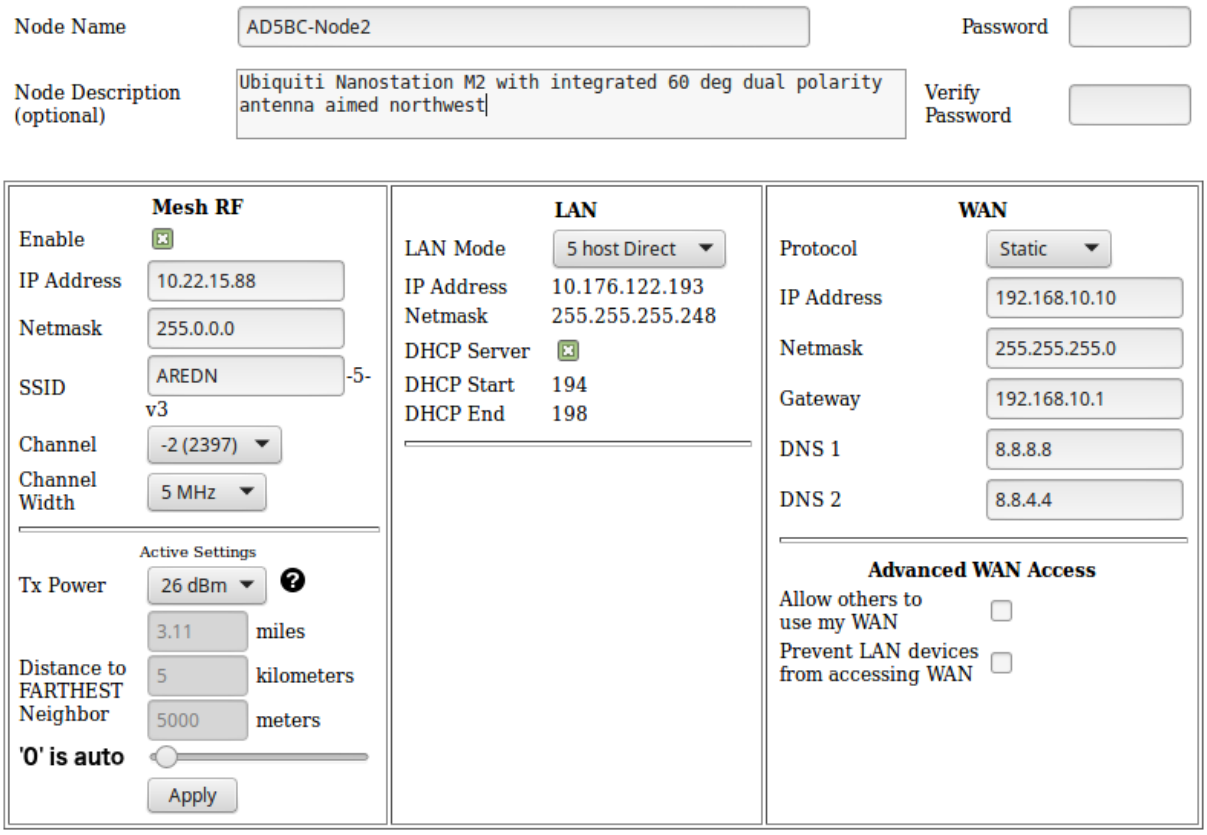
**Reset Values** Click this button to reload the currently saved settings from flash memory, effectively undoing any changes that were made.

**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.




### 8.1.1 Mesh RF Column

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

The **Active Settings** can be adjusted and applied without saving changes or rebooting your node. However, they will return to their original values after a reboot unless you click *Save Changes*. A node may decrease its output power as it increases its data rate in order to maintain a linear spectrum.

**Distance Setting** The *Distance* setting is only applicable to nodes that can communicate directly

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

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

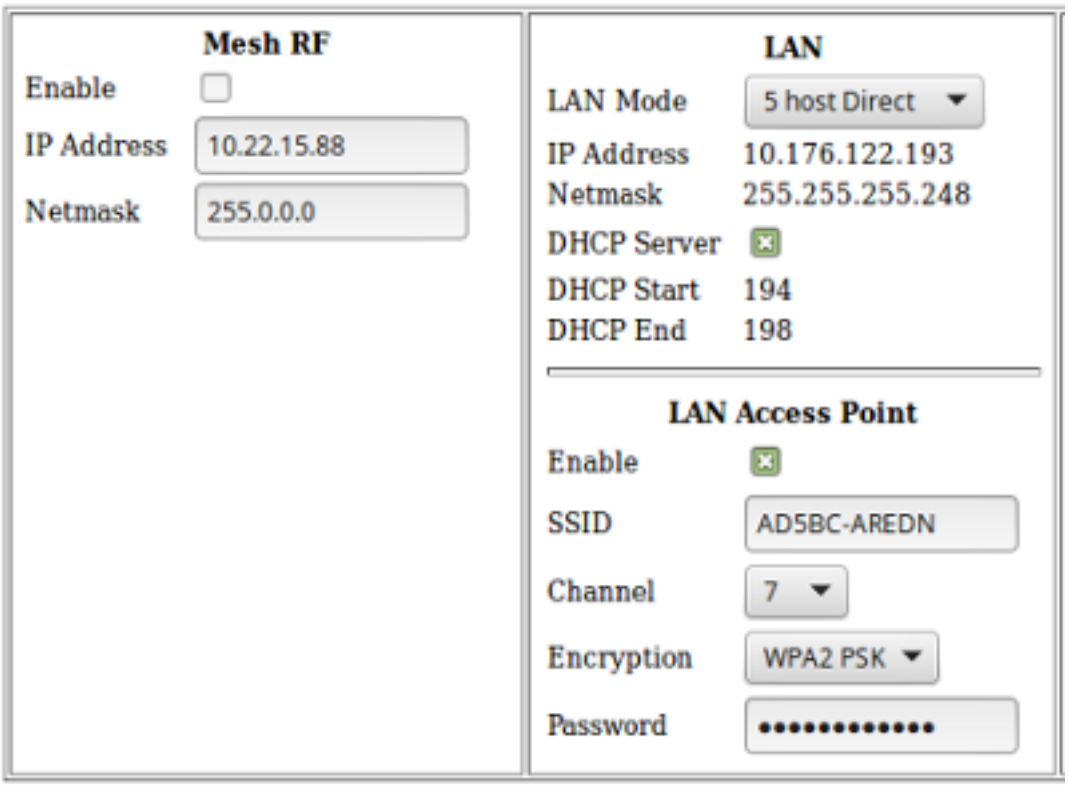
- 20 MHz: 46666 meters
- 10 MHz: 103030 meters
- 5 MHz: 215757 meters

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

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

However, the auto-distance setting does not work well when link quality is marginal or when there are many nodes sharing the channel. In this scenario the round-trip packet timing has a very wide range of values, so the timeout value becomes inflated and inconsistent. Static settings should be used in this situation.

**Enable/Disable Mesh Radio** You can disable your node's radio interface by deselecting the *Enable* checkbox, saving your changes, and rebooting the node. With the Mesh RF interface disabled the *Active Settings* no longer apply and will disappear. Since your node now has an unused RF interface, you will notice that a new section appears which allows you to use the node's radio as an FCC Part 15 *LAN Access Point*. You can enable or disable the LAN AP using the *Enable* checkbox. See the details below for configuring the LAN Access Point.




### 8.1.2 LAN Column

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

The default mode is `5 Host Direct`. In this mode every host on the LAN has direct access to and from the mesh. This mode was created to reduce the amount of manual configuration needed to provide services to the mesh, since many services do not work well if they are hosted behind a NAT (Network Address Translation) router. With *Direct* mode the LAN shares the same address space as the mesh at large. Port forwarding is not needed because NAT is not used, and there is no firewall between the LAN and the mesh.

The mesh address space is automatically managed, so you cannot configure the LAN network settings in *Direct* mode. The only configurable option available in *Direct* mode is the size of the LAN subnet which can accommodate either 1, 5, 13, or 29 LAN hosts. A one host subnet can be used for either a single server or a separate network router using its own NAT which is capable of more advanced routing functions than those available on a mesh node.

It is important not to use a subnet larger than is necessary because the chance of an IP address conflict on the mesh increases with the size of the subnet. The LAN subnet parameters are automatically calculated and depend on the IP address of the *Mesh RF* interface. If a conflict does occur it can be fixed by changing the *Mesh RF* IP address.

The other LAN Mode is `NAT`, and in this mode the LAN is isolated from the mesh. All outgoing

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

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

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

### 8.1.3 WAN Column

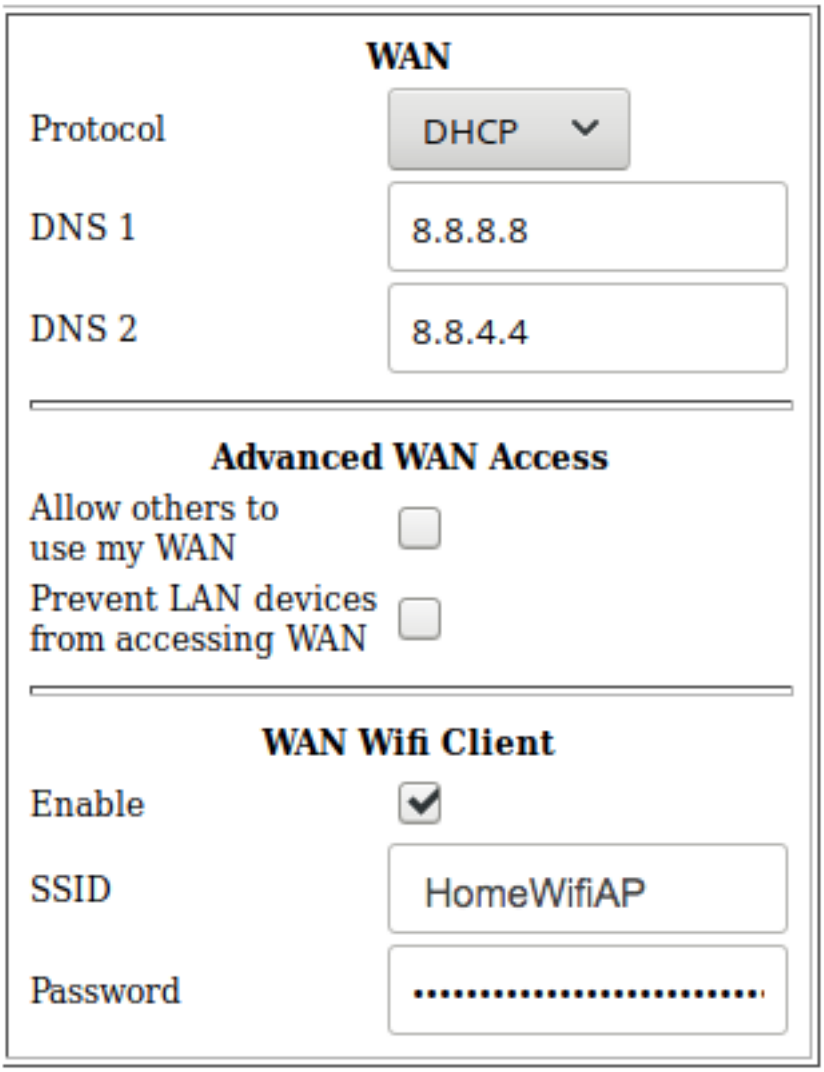
The WAN (Wide Area Network) interface on your node is typically used to connect it to the Internet or to another external network. By default the WAN interface is set to obtain an IP address via DHCP from your upstream network. The DNS (Domain Name System) servers are set by default to use Google's DNS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

If you are not going to use the WAN interface on your node, you can select *disabled* from the *Protocol* dropdown list. If you will be using your node as a *Tunnel Server*, you should reserve an IP address on your router for the node's WAN interface. This will be explained in the *Tunnel Server* section below.

When a node has Internet access on its WAN interface, that access is available to the node itself and to any computers connected via the LAN port. Checking the *Allow others to use my WAN* box will allow this node to route traffic from *all* its interfaces to/from the Internet or other external network. This box is unchecked by default because it is not desirable to route Internet traffic over the radio interface. AREDN® is an FCC Part 97 amateur radio network, so be sure that any traffic which will be sent over the radio complies with FCC Part 97 rules. If you want local wireless Internet access, consider using an FCC Part 15 access point instead of the node's WAN gateway.

The *Prevent LAN devices from accessing WAN* checkbox will tell the node not to advertise that it

can be used as a default gateway. This means that computers on the LAN network will lose their route to the Internet or other networks via your mesh node. This checkbox is deselected by default. If this checkbox is selected your LAN hosts will have no access to the Internet even if your node has Internet access on its WAN interface. You may need to disable the default route if your node needs to be connected to two networks at once, such as being wired to the mesh and connected to a local served agency WiFi network.




If your node has a radio which is not already being used for Mesh RF or as a LAN AP, you can enable it as a WAN interface by checking the *WAN Wifi Client* checkbox. Enter the SSID and authentication string for your wifi AP which has Internet access. The mesh node uses "WPA2 PSK" encryption to connect to the wifi AP. The password length must be a minimum of 8 and maximum of 64 characters. If the key length is 64, it is treated as hex encoded. If the length is 0, then no encryption will be used to connect to an open AP. A single quote character must not be used in the passphrase.

After you have saved changes and rebooted, the node will have Internet access via wifi rather than

requiring a cable plugged into the node's WAN port. In fact, enabling the *WAN Wifi Client* will disable VLAN1, so Internet access will no longer be possible through the physical WAN port.

### 8.1.4 Node VLANs

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

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

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

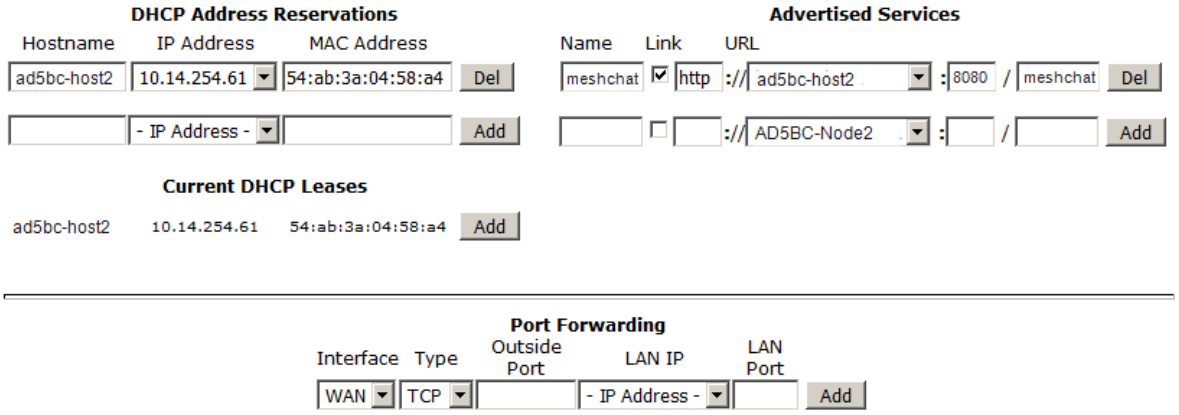
**No VLAN tag** Packets received by the node that are untagged will be identified as LAN traffic from computers on the local area network.

It is important to understand AREDN® VLANs when configuring network smart switches for Internet access, tunneling, or DtD linking of nodes. There are some useful tutorials available on the AREDN® website for configuring VLAN-capable switches: Video or Text+Images. Also, on the AREDN® GitHub site there is more information about node VLANs that have been preconfigured in the firmware images for specific types of radio hardware. For additional information visit this link: Ethernet Port Usage

## 8.2 Port Forwarding, DHCP, and Services

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

---




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

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

Since DHCP leases are dynamic and can change over time, there may be a reason why a host's assigned IP address should be made permanent. This is especially useful if that host will provide an application, program, or service through your node to the mesh network at large. You can permanently reserve that host's DHCP address by clicking the *Add* button 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 your want your advertised service to display an active link in the web browser. This allows mesh users to navigate to your service by clicking the link.

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

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

**Port** Enter the network port on which the service is listening for user connections. There may be several applications provided through a single web server on a node or host using a single port, and in that case a valid application *Path* must be entered after the port number (as in the example above). In other cases the network port alone uniquely identifies the application or program that is listening for user connections to that service. You can click this link for additional information about network ports.

Once you have entered the values for your advertised service, click *Add* to add the service to the **Advertised Services** list. You may also remove an existing advertised service by clicking the *Del* button to delete it from the list.

### 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.

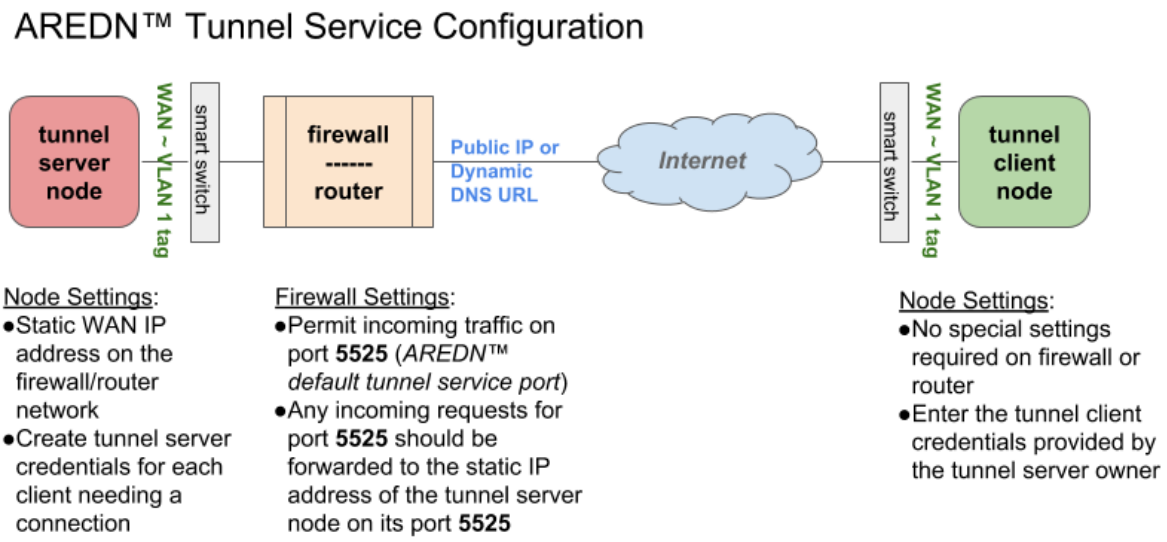
Your node should have Internet access after the smart switch is configured, and you can use the node's new Internet connection to install the *tunneling* software package. This package should be installed on both the tunnel server and the tunnel client nodes.

**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.




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.

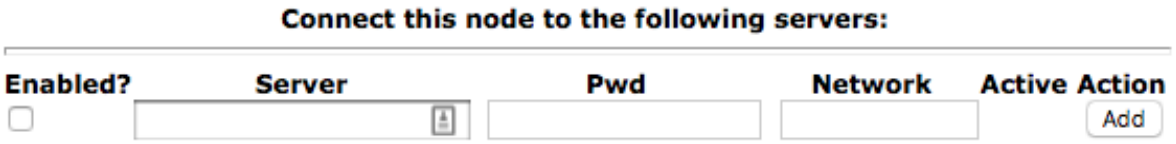
Once these settings are correct, click *Add* to add the new client to the list of authorized tunnel clients. On the right of each entry there is an envelope icon which will automatically open your computer's email program and copy the client settings into a new email which allows you to quickly and easily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

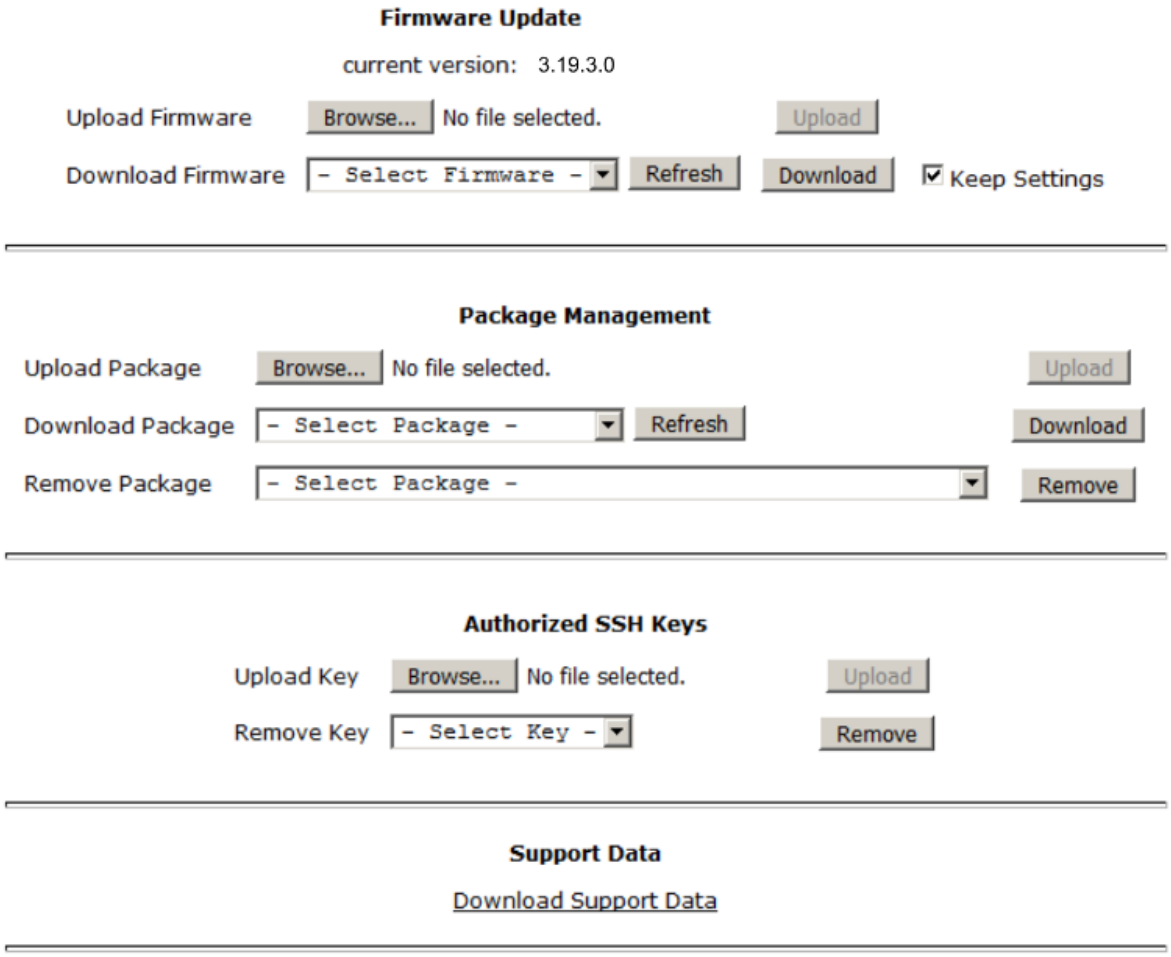
Contact the amateur operator who controls the tunnel server and request client credentials by providing your specific node name. The tunnel server administrator will provide you with the public IP or DDNS (Dynamic Domain Name Service) URL for the tunnel server, the password you are to use, and the network 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.




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.




**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.

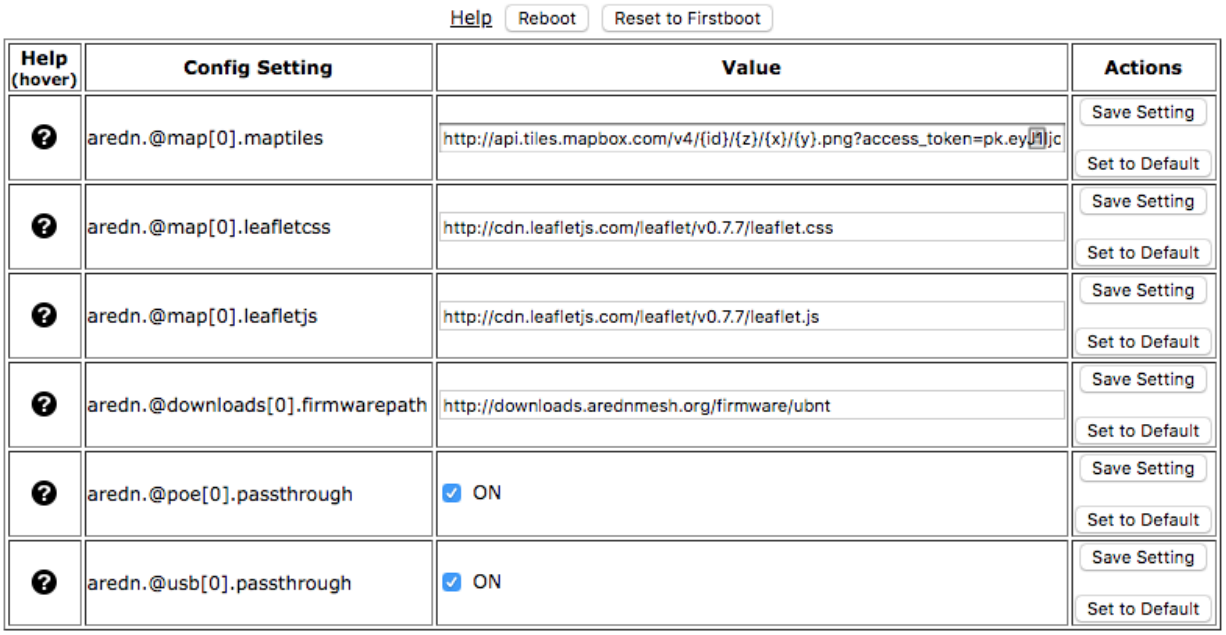
The *Remove Package* list shows all packages currently installed on the node. Selecting a package and clicking *Remove* will uninstall the package. You will only be able to remove packages that you have added. All installed packages are shown, but the pre-installed packages cannot be deleted since they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



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 "N0CALL" 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

## Resumen de redes

Esta ** Guía de diseño de red ** analizará algunos de los principios útiles para crear redes de datos robustas como un servicio tanto para los radioaficionados como para la comunidad en general. Un AREDN | trade | red puede servir como mecanismo de transporte para las aplicaciones en las que las personas confían para comunicarse entre sí en el curso normal de sus interacciones comerciales y sociales, incluido el correo electrónico, el chat, el servicio telefónico, el intercambio de documentos, la videoconferencia y muchos otros programas útiles. . Dependiendo de las características de la implementación, esta red de datos digitales puede operar a velocidades cercanas a Internet con muchas millas entre los nodos de la red.

Hay una variedad de formas de interconectar AREDN | trade | nodos, pero la pregunta más importante que debe responderse es * "¿Cuál es el propósito de esta red en particular?" * Los requisitos específicos de su situación impulsarán el diseño de su red de datos. Por ejemplo, considere los siguientes problemas:

**Temporal o Permanente** ¿Se está desplegando su red como un mecanismo de comunicación a corto plazo, posiblemente para satisfacer las necesidades de un evento de un día o un ejercicio ? Si es así, varios radioaficionados con nodos portátiles pueden establecer rápidamente una red * ad hoc * con un conjunto específico de servicios para satisfacer las necesidades de comunicación para esa situación. Esos nodos y ordenadores probablemente pueden funcionar con baterías portátiles, sin ninguna dependencia de energía externa para un despliegue temporal.

¿Su red está pensada como una infraestructura a largo plazo o permanente para atender las necesidades de comunicación en curso de un área o región local? Si es así, entonces se debe diseñar y construir una topología de red más sofisticada para cumplir con esos requisitos a largo plazo. Puede ser necesario un equipo de radio más robusto o resistente, y se requerirá una alimentación más confiable o recursos de energía renovable fuera de la red para garantizar operaciones consistentes.

**Geografia y terreno** ¿Dónde se necesita la comunicación de datos? ¿Hay ubicaciones específicas donde se requieren nodos de red? ¿Qué nivel de: abbr: cobertura de 'RF (Radio Frequency)' será necesaria para llegar a esos lugares? Los lugares a los que debe llegar la red determinarán el número y la posición de AREDN | trade | nodos

¿Cuáles son las características geográficas del área a través de la cual operará su red de datos? Los diferentes tipos de terreno pueden requerir tipos específicos de conexiones de red para cubrir adecuadamente la región sobre la cual se necesitan comunicaciones de datos. Un terreno más exigente puede requerir una mayor cantidad de nodos intermedios o posiblemente sistemas de antena y estructuras de montaje de mayor ganancia.

**Expanción y crecimiento de la red ** ¿Su red necesitará expandirse o adaptarse a las condiciones cambiantes con el tiempo? Las redes de malla son ideales para el crecimiento * ad hoc * y el enrutamiento de menor costo basado en la disponibilidad de nodos. Sin embargo, a medida que se agregan más dispositivos a la red, la topología se vuelve más complicada y el tráfico de datos se puede enrutar a través de múltiples "saltos" para llegar a su destino previsto. Esto podría resultar en una mayor latencia en la red, con algunos segmentos de red que se vuelven casi inutilizables si no se pueden alcanzar los umbrales de tiempo de respuesta de la aplicación.

**Aplicaciones y rendimiento** ¿Qué programas, aplicaciones o servicios de red se deben proporcionar para cumplir el propósito de la red? Cada aplicación generará una cierta cantidad de tráfico de datos, y algunos programas o servicios requieren más datos que otros. La red necesita ser diseñada para pasar adecuadamente el tráfico para las aplicaciones requeridas.

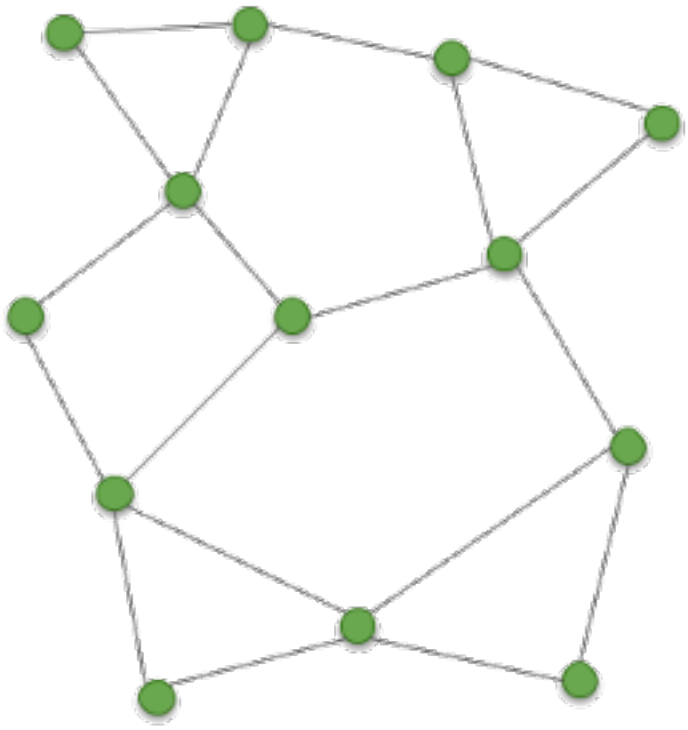
¿Cuántos usuarios simultáneos generarán tráfico en la red y en diferentes momentos? A medida que aumenta el número de usuarios, también aumentará la cantidad de datos que atraviesan la red. Además, con un número creciente de nodos en la red, habrá un aumento correspondiente en la cantidad de *OLSR (protocolo de enrutamiento de estado de enlace optimizado) <https://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol>* _ tráfico que es necesario para mantener la red de malla. Un AREDN | trade | red debe estar diseñada para manejar la carga de trabajo esperada.

Con estos factores en mente, siempre es mejor mantener la red lo más simple posible e incluir solo aquellos servicios que se requieren. Asegúrese de diseñar su red para que cumpla su misión y se adapte a su propósito.

# CHAPTER 10

# Network Topologies

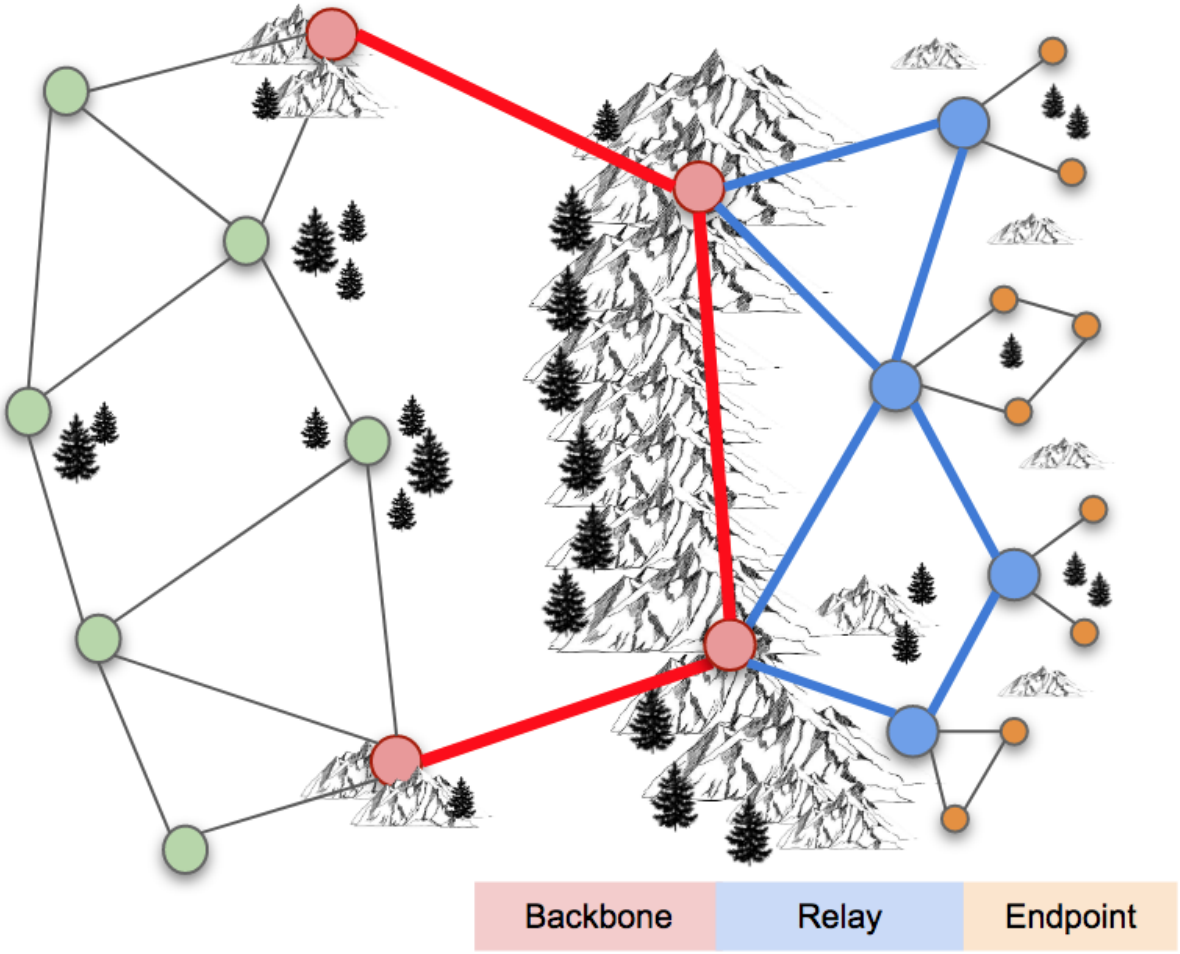
Every AREDN® node is capable of automatically joining an AREDN® mesh network which is operating with the same SSID, channel, and bandwidth. A *Mesh* topology consists of independent nodes which each explore their surroundings by broadcasting their identity and listening for their neighbors' responses. 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 network 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.

**Relay Links** Relay 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 "intermediate" nodes. They are usually installed on medium-height towers or buildings in order to achieve high signal quality with good line of sight to other relay 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.

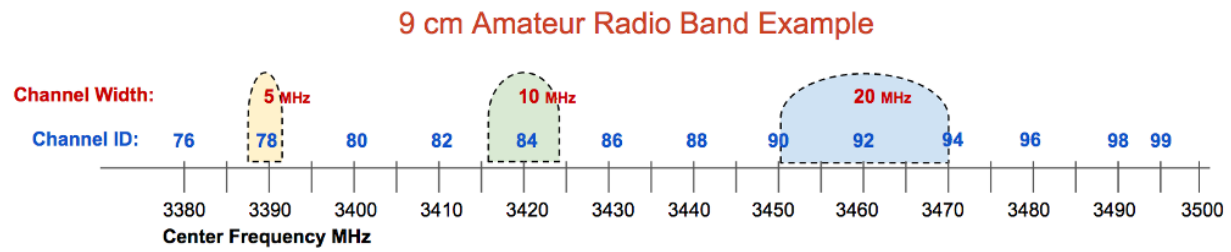
Different types of radio links may be needed to connect all of the mesh nodes that are required in order to fulfill the purposes for your network. The ultimate goal is to have a reliable data network that accomplishes its purpose for providing services to the intended destinations and users.

# CHAPTER 11

# Características del espectro radioeléctrico

AREDN | trade | redes operan en el espectro de radio de microondas y los radioaficionados con autorización especial tienen acceso exclusivo a muchas de estas frecuencias. Para las bandas en las que los radioaficionados comparten el espectro, existe una mayor probabilidad de interferencia de RF que puede hacer que algunas frecuencias sean inutilizables para AREDN | trade | redes de datos. Para obtener los mejores resultados, seleccione frecuencias que no se comparten con otros usuarios sin licencia.

Cada banda se divide en canales, cada uno de los cuales consiste en un desplazamiento de frecuencia de 5 MHz identificado por la frecuencia central del canal y se le asigna una etiqueta numérica. En el siguiente ejemplo, puede ver que un canal seleccionado puede usar más o menos el rango de frecuencia en función del ancho del canal elegido. Cuanto más ancho sea el canal, más superposición habrá con los canales adyacentes. Los canales anchos tienen el efecto de reducir la cantidad de canales que no se superponen o que no interfieren y que estarán disponibles para su uso. Al seleccionar canales y anchos de banda, asegúrese de usar canales que no se superpongan. Los dispositivos que no usan el mismo canal y ancho de canal que se superponen interferirán entre sí y no pueden comunicarse para coordinar el uso compartido del ancho de banda.




Toda la banda de 33 cm se comparte con otros usuarios autorizados de la FCC. Todos los canales

superiores en la banda de 13 cm se comparten con los usuarios estándar de la FCC Parte 15: abbr: *WiFi (IEEE 802.11x)*, al igual que todos los canales inferiores en la banda de 5 cm. El único rango de frecuencia que los radioaficionados no comparten actualmente con usuarios sin licencia es la banda de 9 cm, en la cual el ejército de los EE. UU. puede operar ocasionalmente con radiobalizas. La siguiente tabla enumera cada banda de radioaficionados, rango de frecuencia, ancho de banda total de asignación y la cantidad de canales disponibles para AREDN | trade | redes.

| Banda | Frecuencia | Ancho de banda | Canales |
|---|---|---|---|
| 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 |

La tabla anterior muestra que la banda de 9 cm tiene el mayor ancho de banda disponible en sus canales no compartidos, mientras que la banda de 5 cm tiene la siguiente mayor cantidad de ancho de banda disponible en canales no compartidos. La elección de una banda de frecuencia para conexiones AREDN | trade | red depende de varios factores diferentes, pero puede "mezclar y combinar" bandas en el diseño de su red siempre que ambos lados de un enlace de radio usen la misma banda, canal y ancho de canal.

Tiene la opción de seleccionar el ancho del canal para cada enlace. Cuando utilice canales en la parte superior o inferior de una banda, asegúrese de que el ancho elegido no se transmitirá fuera de la asignación de la Parte 97 de la FCC para esa banda. Diferentes anchos de canal pueden producir un mejor rendimiento que otros. En algunas áreas, los operadores usan diferentes canales para aislar enlaces, por lo que pueden necesitar usar canales de 10 MHz en lugar de 20 MHz para asegurarse de tener suficientes canales disponibles. Además, los enlaces de larga distancia simplemente tienen un mejor rendimiento con anchos de canal de 10 MHz frente a 20 MHz o 5 MHz. Pruebe el rendimiento de sus enlaces utilizando varios anchos de canal para asegurarse de que estén optimizados.

Algunas de las ventajas y desventajas de cada rango de frecuencia se explican en las siguientes secciones.

## 11.1 Características de la banda de 900 MHz

** Desventajas ** Toda la banda de 33 cm se comparte entre varios servicios de radio autorizados por la FCC. La desventaja de usar esta banda para conexiones de red AREDN | trade | es que en todas las áreas remotas, excepto en las más remotas, el nivel de ruido de RF puede ser muy alto, lo que reduce el: abbr: *SNR (relación señal / ruido)* y da como resultado la pérdida de paquetes, retrasos en la retransmisión y una menor calidad de enlace utilizable.

Otra desventaja es que el equipo puede ser más costoso que los dispositivos que operan en las bandas de 2.4 y 5.8 GHz. Además, toda la banda es bastante estrecha (25 MHz), lo que significa

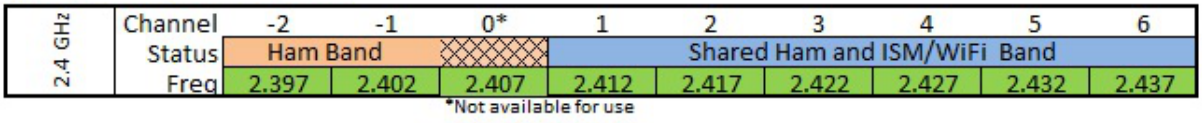
que solo pueden existir uno, dos o cinco canales de radio en este rango de frecuencia compartida, dependiendo del ancho del canal seleccionado.

**Ventajas** La ventaja de esta banda de frecuencia es que su longitud de onda más larga la hace más adecuada para penetrar algunos tipos de obstrucciones que normalmente bloquearían las señales a frecuencias más altas. Sus características de propagación: abbr: *NLOS (Non Line of Sight)* pueden ser exactamente lo que se necesita para establecer un enlace de RF entre dos ubicaciones difíciles.

## 11.2 Características de la banda de 2.4 GHz

**Desventajas** Los canales superiores de la banda de 13 cm se comparten con varios otros servicios autorizados por la FCC. Dependiendo de las condiciones locales de RF, puede que no sea posible utilizar estos canales compartidos debido al alto nivel de ruido que reduce: abbr: *SNR (relación señal / ruido)* y disminuye la calidad de la señal. Esto deja a los radioaficionados solo con dos canales no compartidos con un posible ancho de banda de 10 MHz cada uno.

Una preocupación con todas las bandas de frecuencias más altas es que debe haber una línea de visión clara entre los nodos a cada lado del enlace. Esto significa que no solo los nodos necesitan tener una ruta directa sin obstrucciones, sino que la Zona Fresnel entre los nodos también debe estar despejada. El diámetro de la zona de Fresnel depende de la frecuencia y la distancia entre los nodos. Por ejemplo, en un enlace en la banda de 13 cm con 10 millas entre nodos, el primer radio de la Zona Fresnel será de 72 pies. Si menos del 20% de la zona de Fresnel está obstruida, hay poca pérdida de señal, pero cualquier bloqueo más allá del 40% causará una pérdida de señal significativa y podría inutilizar el camino. En la banda de 13 cm, el radio del 60% sin bloqueo es de aproximadamente 43 pies, que a menudo es más alto que la mayoría de los nodos * Intermedio * o * Última milla * que se han instalado. Se debe considerar cuidadosamente la altura del nodo y el terreno entre los nodos para minimizar las obstrucciones.




**Ventajas** Dentro del rango de frecuencia disponible, tiene la opción de seleccionar anchos de canal de 5, 10 o 20 MHz. Un ancho de canal mayor proporcionará velocidades de datos más altas. Sin embargo, un efecto de reducir el ancho del canal es aumentar: abbr: *SNR (relación señal / ruido)* para mejorar la calidad de la señal. Por ejemplo, cambiar de un ancho de canal de 20 MHz a 10 MHz dará como resultado una ganancia de señal de 3 dB y podría marcar la diferencia entre un enlace marginal y uno utilizable. Solo recuerde que cuando corta el ancho del canal a la mitad, también reduce su rendimiento máximo a la mitad. Pruebe cuidadosamente sus enlaces para garantizar un rendimiento óptimo.

Una ventaja para la banda de 13 cm es que los equipos de radio y los sistemas de antena están más disponibles y son menos costosos debido a la mayor demanda del consumidor. Existe una gran variedad de equipos de varios fabricantes que respaldan AREDN |trade | y operan en esta

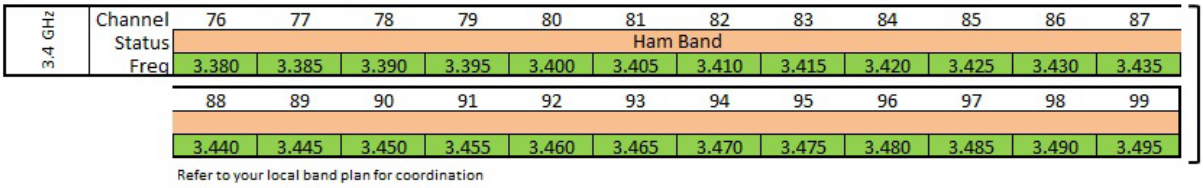
banda. Con una línea de visión clara y antenas bien sintonizadas, las señales de 2.4 GHz pueden propagarse a través de distancias muy largas.

Características de la banda de 3.4 GHz

————————-Características————————-

**Desventajas** Como se mencionó anteriormente, debe haber una línea de visión clara y la zona de Fresnel entre los nodos también debe estar despejada. Para un enlace en la banda de 9 cm con 10 millas entre nodos, el primer radio de la zona de Fresnel será de 62 pies, que es menor que la banda de 13 cm discutida anteriormente. Sin embargo, el radio del 60% sin bloqueo sigue siendo de unos 37 pies. Considere el nodo: abbr: *AGL (altura sobre el nivel del suelo)* y el terreno para minimizar las obstrucciones.

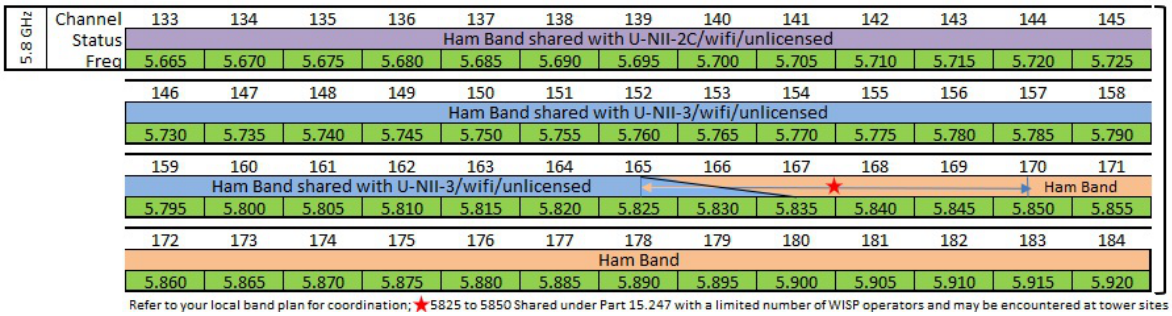
El equipo para la banda de 9 cm está menos disponible y generalmente es más costoso debido a la menor demanda del consumidor. Se debe tener cuidado al seleccionar radios para no confundirlos con los dispositivos más comunes: abbr: *WiMAX (IEEE 802.16)* que están diseñados para el rango de 3,65 GHz y no son compatibles con AREDN | trade | firmware.




**Ventajas** La principal ventaja de usar la banda de 9 cm es que tiene más ancho de banda disponible para usar en canales no compartidos que cualquier otra banda. Puede seleccionar anchos de canal de 5, 10 o 20 MHz, con anchos de canal más grandes que proporcionan velocidades de datos más altas. Recuerde que reducir el ancho del canal aumentará: abbr: *SNR (relación señal / ruido)* para mejorar la calidad de la señal si eso es un problema para un enlace en particular. El equipo en la banda de 9 cm es adecuado para * Backbone Links * ya que hay pocas posibilidades de interferencia de otros dispositivos que comparten estas frecuencias en los sitios de la torre. Con una línea de visión clara y antenas bien sintonizadas, las señales de 3,4 GHz pueden propagarse a través de distancias muy largas.

## 11.3 Características de la banda de 5.8 GHz

**Desventajas** Como se mencionó anteriormente, debe haber una línea de visión clara y la zona de Fresnel entre los nodos también debe estar despejada. Para un enlace en la banda de 5 cm con 10 millas entre nodos, el primer radio de la Zona Fresnel será de 46 pies, que es mucho menor que las bandas de frecuencia discutidas anteriormente. Sin embargo, el radio del 60% sin bloqueo en la banda de 5 cm todavía es de aproximadamente 28 pies. Asegúrese de tener en cuenta el nodo: abbr: *AGL (altura sobre el nivel del suelo)* y el terreno para lograr una línea de visión clara entre los nodos.




**Ventajas** Una ventaja de usar la banda de 5 cm es que contiene 52 canales, y muchos de ellos en el extremo superior de la banda se subutilizan con menos posibilidades de interferencia. Puede elegir anchos de canal de 5, 10 o 20 MHz, canales con mas ancho proporcionan velocidades de datos más altas. Recuerde que la reducción del ancho del canal aumentará: abbr: *SNR (Relación señal / ruido)* para mejorar la calidad de la señal si eso es un problema para un enlace problemático.

El equipo de radio y los sistemas de antena para esta banda están fácilmente disponibles y son menos costosos debido a la mayor demanda del consumidor. Existe una gran variedad de equipos de varios fabricantes que respaldan el AREDN |trade | firmware y operan a través de los 52 canales disponibles. Los sistemas de radio y antena para esta banda que son similares en tamaño a los de otras bandas y a menudo tendrán mayor ganancia. Los dispositivos en la banda de 5 cm también son adecuados para * Backbone Links * ya que hay pocas posibilidades de interferencia de RF de otras radios que comparten estas frecuencias en sitios de alto perfil. Con una línea de visión clara y antenas bien sintonizadas, las señales de 5.8 GHz pueden propagarse a través de distancias muy largas.

Hay diferentes rangos de frecuencia disponibles para conectar los nodos de malla necesarios para cumplir con los propósitos de su red. Al planificar las frecuencias que se desplegarán en ubicaciones específicas, puede ser útil usar un * analizador de espectro * para identificar los rangos que ya están en uso. El objetivo final es tener una red de datos confiable que cumpla su propósito de proporcionar servicios a los destinos y usuarios previstos.

# CHAPTER 12

# Planificación de canales

La sección anterior identificó los diferentes canales en cada banda de frecuencia que están disponibles para AREDN | trade | redes. Los dispositivos a cada lado de un enlace de radio deben usar la misma banda de frecuencia, canal, ancho de canal y SSID. Sin embargo, más allá de ese requisito, tiene bastante flexibilidad para seleccionar los canales de radio que asegurarán la mayor calidad de señal y rendimiento para su red. En un AREDN | trade | red con varios nodos distribuidos en un área geográfica limitada, todos los nodos pueden usar la misma banda, canal y ancho de canal. Esto les permite establecer la red de malla y enrutar datos a cualquiera de los sitios según sea necesario.

Sin embargo, a medida que más nodos se unen a la red o cuando varios nodos son: abbr: *colocados (mismo sitio físico)* y comparten el mismo canal, es posible que el rendimiento general de la red se degrade. Para un AREDN | trade | red para ampliar en tamaño y complejidad, la coordinación de frecuencias y la planificación de canales se vuelven cada vez más importantes. Para planificar el crecimiento futuro, los grupos de malla pueden necesitar dividir el tráfico de la red mediante la asignación de canales para áreas específicas o tipos de enlaces con el fin de garantizar que la red pueda soportar los servicios esperados.

## 12.1 operación de red inalámbrica

Una red inalámbrica es un medio semidúplex compartido en el que solo debe transmitir una estación a la vez. En ese sentido, las operaciones inalámbricas son análogas a otros tipos de transmisiones de radio. Si dos estaciones conectan sus transmisores al mismo tiempo, interferirán entre sí en la medida en que ninguna de ellas reciba el mensaje de la otra. Es por eso que se implementan procedimientos de control de red para garantizar el acceso controlado a un canal de radio

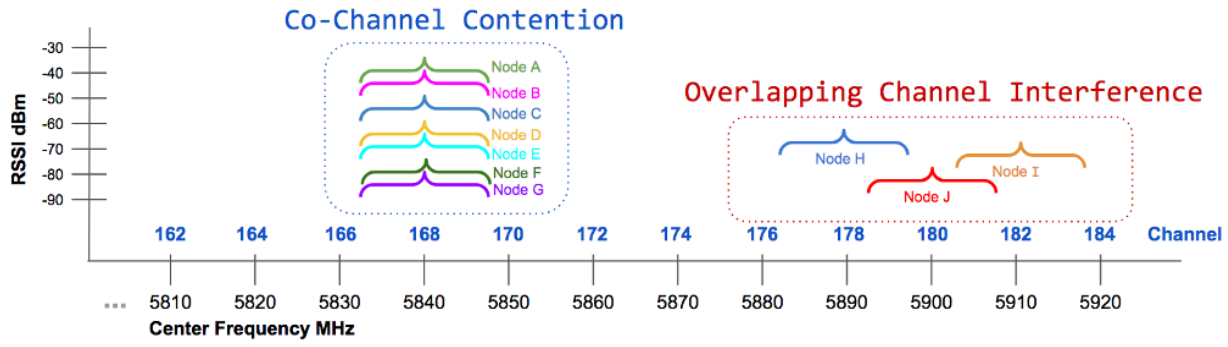
durante la comunicación de emergencia.

AREDN | trade | firmware verifica automáticamente el acceso de la estación al medio inalámbrico mediante la implementación de *IEEE 802.11a / b / g / n <https://en.wikipedia.org/wiki/IEEE_802.11n-2009>* _ standards y *Carrier Sense Multiple Access / Evitación de colisiones (CSMA / CA) <https://en.wikipedia.org/wiki/Carrier-sense_multiple_access> '_. Esta tecnología de escuchar antes de hablar ayuda a los nodos a determinar si un canal está ocupado. Cada nodo realiza una * Clear Channel Assessment (CCA) * además de usar 'Request to Send / Clear to Send (RTS / CTS) <https://en.wikipedia.org/wiki/IEEE_802.11_RTS/CTS>* _ mensajes para negociar el acceso a un canal. También se requiere una cantidad insignificante de tráfico de red para *OLSR (protocolo de enrutamiento de estado de enlace optimizado) <https://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol>* _ para mantener rutas para la red de malla en su conjunto, pero ese OLSR tráfico es una fracción muy pequeña del total.

En una red inalámbrica de un solo canal, cualquier nodo que necesite transmitir se coordinará automáticamente con los otros nodos para obtener un canal libre. Esto es así por diseño, pero a medida que más dispositivos intentan obtener acceso al mismo canal, existe un mayor potencial para que cada nodo espere más tiempo para poder transmitir. Esto puede resultar en una mayor latencia y una disminución del rendimiento de la red a medida que aumenta el número de nodos de red.

### 12.1.1 canal de contención

El concepto de * Interferencia de canales superpuestos * se ilustra en el lado derecho del siguiente diagrama de exploración de canales. * La interferencia de canales superpuestos * es muy grave, pero se puede eliminar seleccionando canales no superpuestos para todos los dispositivos que acceden a su red de malla. Un segundo problema relacionado con el funcionamiento de las redes inalámbricas se ilustra en el lado izquierdo del diagrama. Se denomina comúnmente * Interferencia cocanal *, pero se describe con mayor precisión como * Contención cocanal * o * Cooperación cocanal *.




En este ejemplo, varios nodos deben compartir un solo canal, por lo que todos negocian la oportunidad de transmitir. Cualquier nodo que necesite transmitir utilizará la tecnología de escuchar

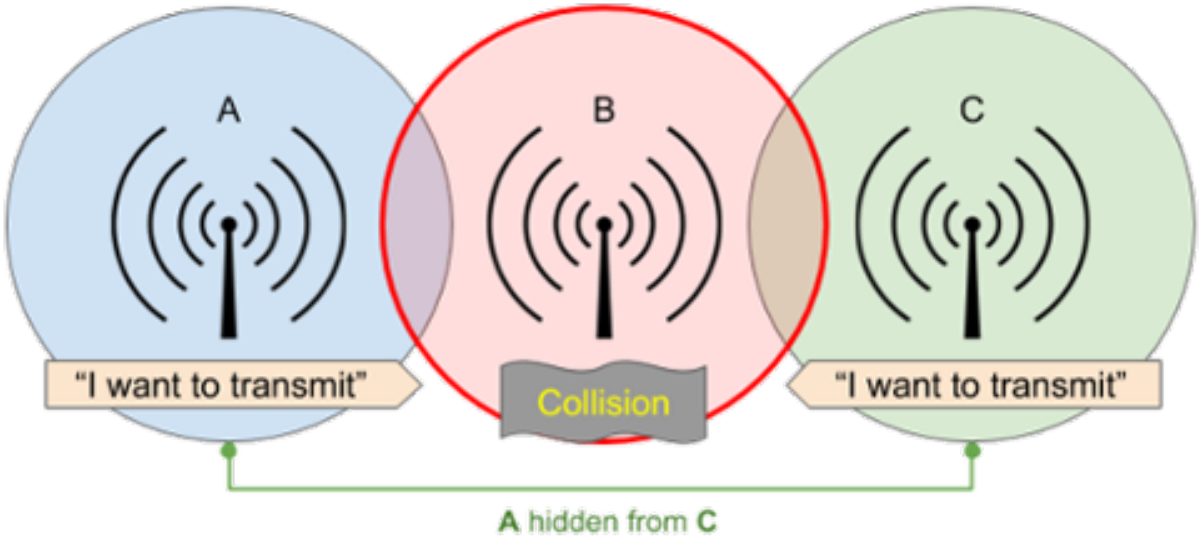
antes de hablar para determinar si el medio está ocupado. Si el canal parece libre, el nodo intentará transmitir datos. Si el canal está ocupado, el nodo diferirá la transmisión hasta que el canal parezca despejado. En una red de alta densidad donde una gran cantidad de nodos comparten un solo canal, los procesos normales de negociación pueden provocar una degradación significativa del rendimiento. Desde la perspectiva del usuario final, un canal sobrecargado puede hacer que la red parezca lenta o incluso inutilizable.

Este ejemplo no pretende mostrar teniendo solo siete nodos se va sobrecargar un canal. No existe una regla general establecida para compartir canales que especifique cuántos nodos son demasiados. La respuesta depende del número de nodos, el ancho de banda en uso para soportar los servicios requeridos, las cualidades de la señal de enlace y otras características de la red. En función de estos parámetros, un canal compartido puede funcionar bien con muchas docenas de nodos, mientras que otra red puede ver una degradación del rendimiento con una cantidad significativamente menos nodos que comparten un mismo canal. Muchos factores interactúan para influir en el rendimiento de la red, pero pronto será obvio para los usuarios si la red se comporta como se esperaba.

Hay varias herramientas disponibles para probar el rendimiento de la red, como * ping * para medir la latencia, * traceroute * para identificar cómo se enruta el tráfico y * iperf3 * para estimar el rendimiento de la red. Las mediciones periódicas junto con las percepciones del usuario pueden ser útiles para determinar si la separación de canales sería beneficiosa. Es un subproducto esperado de cómo funcionan normalmente las redes inalámbricas, pero el rendimiento puede mejorarse al planificar los canales asignados para sus dispositivos de malla como se describe en la sección ** Planes de canales ** a continuación.

### 12.1.2 Nodos ocultos

En cualquier red inalámbrica habrá nodos que no estén dentro del alcance de radio entre sí pero que compartan el mismo canal. En el diagrama de ejemplo, ** A ** puede escuchar ** B ** pero no puede escuchar ** C **. Como ** A ** y ** C ** están *ocultos el uno del otro <https://en.wikipedia.org/wiki/Hidden_node_problem> _*, pueden intentar transmitir en el canal compartido al mismo tiempo sin sabiéndolo. Debido a sus ubicaciones relativas y a cualquier retraso de red asociado, puede parecer que cada nodo tiene un canal libre para la transmisión.




*Solicitud de envío / Borrar para enviar (RTS / CTS) <https://en.wikipedia.org/wiki/IEEE_802.11_RTS/CTS>* _ los mensajes son utilizados por AREDN | trade | nodos para minimizar o eliminar este problema. Por ejemplo, el nodo ** A ** transmite un breve mensaje RTS con un intervalo de tiempo / duración propuesto para transmitir su flujo de datos completo. El nodo ** B ** recibe esa solicitud y transmite un CTS para ese intervalo de tiempo. El nodo ** C ** no pudo escuchar el RTS original, pero escuchará el mensaje CTS y diferirá sus transmisiones durante ese intervalo de tiempo.

Otros dos enfoques también pueden aliviar el problema del nodo oculto. Es posible que pueda hacer que los nodos ocultos sean visibles entre sí, por ejemplo, aumentando la intensidad de la señal. La alternativa es aislar completamente los nodos colocándolos en diferentes bandas o canales. Dado que los nodos que usan antenas direccionales son casi invisibles para otros que no están ubicados en el haz de la antena, las antenas direccionales deben usarse con cuidado al compartir un canal. Puede ser más apropiado crear un enlace separado entre los sitios y colocar las radios en una banda o canal diferente.

Otro caso es cuando hay un enlace de baja calidad sobre el cual se debe enrutar todo el tráfico. El apretón de manos y las retransmisiones de datos pueden hacer que todos los demás nodos esperen. Toda la red puede verse afectada por una ruta de baja calidad que se convierte en un solo cuello de botella. Si es posible, debe aumentar la calidad de la señal de ese enlace vital o instalar un mejor enlace como una ruta alternativa.

### 12.1.3 Aleteo de ruta

Este es otro problema que puede conducir a problemas de rendimiento en una red. Puede tener rutas paralelas entre dos * Nodos remotos *, y estas rutas tienen valores similares: abbr: *ETX (métrica de transmisión esperada)* que indica que el costo de usar cualquiera de las rutas es comparable. Puede parecer que estos dos caminos funcionan bien la mayor parte del tiempo.

Sin embargo, cuando una aplicación de uso intensivo de ancho de banda, como la videoconferencia,

comienza a enviar tráfico a través de una de las rutas, puede notar que el enlace se atasca y el: abbr: *ETX (métrica de transmisión esperada)* caerá por debajo de la otra ruta . En este punto: abbr: *OLSR (protocolo de enrutamiento de estado de enlace optimizado)* cambia a la ruta alternativa que ahora tiene un costo más bajo. Luego, la transmisión de video empantana su nueva ruta, lo que reduce: abbr: *ETX (métrica de transmisión esperada)*, y: abbr: *OLSR (protocolo de enrutamiento de estado de enlace optimizado)* vuelve al enlace original cuyo: abbr: ‘ ETX (métrica de transmisión esperada) ‘ha mejorado. Esta situación puede continuar indefinidamente, sin que ninguna ruta pueda entregar el tráfico adecuadamente.

Este problema puede ocurrir en enlaces de múltiples saltos con similar: abbr: *ETX (métrica de transmisión esperada)* que parece funcionar bien hasta que se cargan con tráfico. Luego, comienza a ocurrir la pérdida de paquetes, las conexiones caducan y ninguna de las rutas es confiable durante ese ciclo. Una solución podría ser mejorar la carga del enlace de múltiples saltos aumentando la calidad de la señal de los enlaces a lo largo de una de las rutas. Por el contrario, también puede reducir la potencia en la ruta alternativa para aumentar su costo. Si se debe pasar tráfico de ancho de banda intensivo entre dos puntos finales remotos, el mejor enfoque sería diseñar una ruta más sólida entre esos dos puntos finales para satisfacer esa necesidad.

## 12.2 Planes de canales y coordinación de frecuencia

Puede experimentar un rendimiento deficiente de la red si hay demasiados nodos que usan la misma banda y canal. Aquí hay un ejemplo simple para ilustrar el problema: una ruta de tres saltos desde QTH1 a Tower1 a Tower2 a QTH2. Si todos los enlaces usan el mismo canal, solo un nodo a la vez puede enviar datos. Esto reduce instantáneamente el rendimiento en un tercio o más y aumenta la latencia con la sobrecarga del protocolo. Para mejorar el rendimiento, puede configurar cada enlace para usar un canal diferente, permitiendo transmisiones simultáneas. Por ejemplo, los nodos de la torre colocada podrían vincularse DtD a través de Ethernet, con QTH1 y Tower1 usando el canal 172 de 5 GHz, mientras que QTH2 y Tower2 usan el canal 176. Antes de implementar este plan de canales, es posible tener un flujo de video HD y una llamada VoIP con frecuentes abandonos. Después de implementar el plan de canales, debería ser posible tener tres transmisiones de video HD y varias llamadas VoIP simultáneamente con pocos abandonos.

Dependiendo de la banda de frecuencia que esté utilizando, hay diferentes opciones disponibles para asignar canales no compartidos no superpuestos a sus dispositivos de malla. Como se muestra en el cuadro a continuación, en la banda de 2,4 GHz que utiliza un ancho de canal de 5 MHz, solo hay dos canales que no se superponen y que aún no se comparten con otros usuarios sin licencia. En la banda de 3.4 GHz que usa los canales pares de 10 MHz, hay once canales no superpuestos. En la banda de 5.8 GHz que utiliza canales pares de 10 MHz, hay 25 canales no superpuestos, pero solo ocho de ellos no se comparten con otros usuarios potenciales del espectro.

Idealmente, las zonas de cobertura de RF (a veces llamadas “celdas”) deberían usar canales diferentes. La cobertura de celdas superpuestas puede proporcionar una conectividad más amplia, pero las zonas de cobertura superpuestas no deben usar frecuencias de RF superpuestas. .. image:: _images/channel-reuse-example.png

**alt** Example Channel Reuse Plan

**align** center

El mapa de cobertura de ejemplo muestra que se han asignado cuatro canales diferentes para lograr una cobertura amplia al segmentar áreas específicas en zonas para reducir la contención cocanal. Cabe señalar que incluso un plan de reutilización de canales como este puede no eliminar todas las instancias de contención. Por ejemplo, si un nodo está en los bordes exteriores de una zona de cobertura o se eleva muy por encima del nivel del suelo, sus transmisiones pueden propagarse a una celda distante utilizando el mismo canal. Las radios en la otra celda diferirán si escuchan las transmisiones del nodo original, aunque se originen en una celda diferente. Puede ser necesario cierto grado de experimentación para minimizar la contención y maximizar el rendimiento de la red.

## 12.3 Nodos que comparten ubicación

En algunos sitios puede haber varios dispositivos montados en el mismo edificio o estructura. La foto de la derecha muestra muchos nodos colocados en una sola torre. La degradación del rendimiento de la red puede ocurrir si estos nodos comparten una banda y canal de RF. Por ejemplo, cuando dos antenas sectoriales se colocan y comparten el mismo canal, el rendimiento de la red para ese sitio se reducirá a la mitad o más. Si tiene nodos colocados, entonces tiene sentido permitir que los dispositivos pasen tráfico a través de su interfaz Ethernet (como se describe a continuación) en lugar de obligarlos a usar su canal de radio.

### 12.3.1 Enlace de dispositivo a dispositivo (DtD)

En su configuración más básica para dos nodos colocados, se conecta un cable Ethernet entre el puerto PoE * LAN * de cada dispositivo. : abbr: *OLSR (Protocolo de enrutamiento de estado de enlace optimizado)* asignará un "costo de enlace" muy bajo (0.1) a la conexión DtD y enrutará automáticamente el tráfico entre los nodos a través de Ethernet en lugar de hacer que el canal de RF esté ocupado.

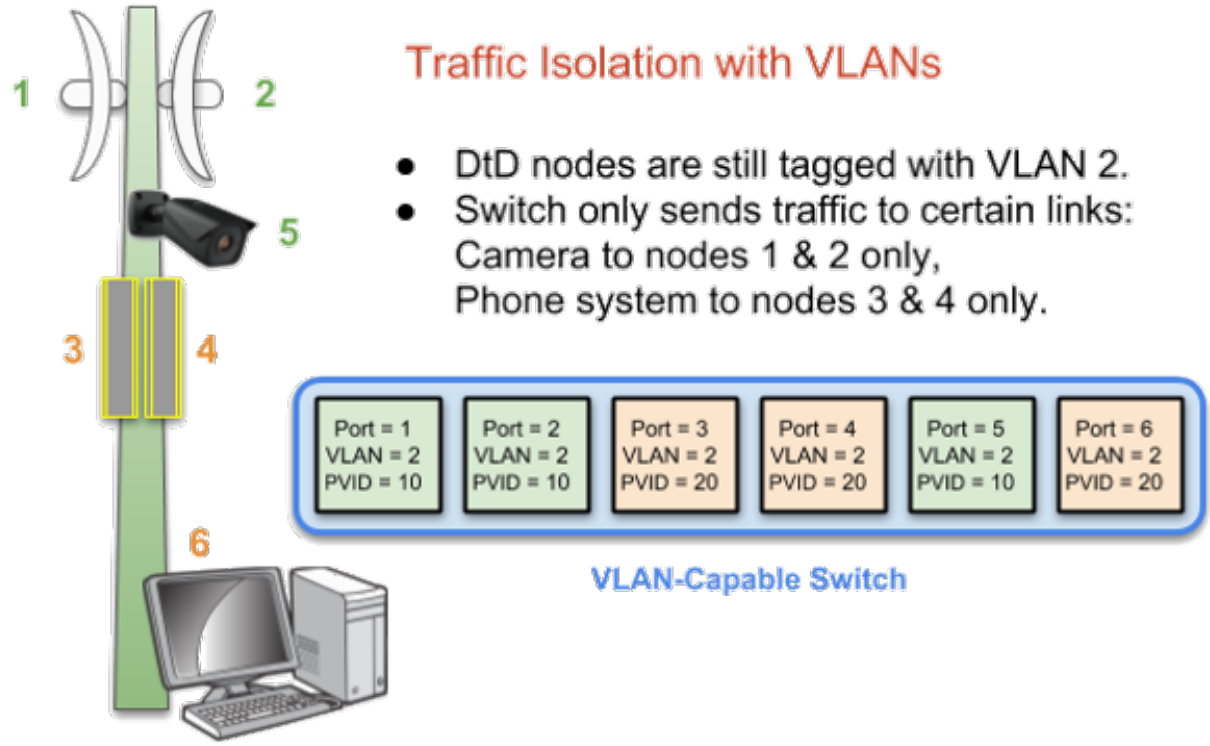
Una ventaja adicional de la vinculación DtD es que puede vincular nodos que operan en diferentes bandas y canales. Los nodos que usan * Separación de canales * para segmentar el tráfico aún pueden pasar datos a altas velocidades a través de su enlace DtD y ser miembros de una sola red. En un sitio de torre como el que se muestra aquí, puede vincular nodos de 2,4 GHz, 3,4 GHz y 5,8 GHz a la misma red. De hecho, en un sitio concurrido como este, es una buena práctica utilizar el enlace DtD, porque de lo contrario la contención del canal de RF podría inutilizar la red.

Idealmente, debe configurar sus nodos coubicados para usar diferentes bandas y canales, y luego configurar enlaces DtD entre los nodos para garantizar que el tráfico se enrute de manera eficiente sin generar embotellamientos de RF o retrasos. : abbr: *OLSR (Protocolo de enrutamiento de estado de enlace optimizado)* siempre elegirá primero la ruta DtD al pasar el tráfico entre los nodos vinculados. Cada AREDN | trade | El nodo reconoce los paquetes entrantes etiquetados con: abbr: *VLAN (Red de área local virtual)* 2 como tráfico DtD. .. image:: _images/dtd-linking.png

**alt** DtD Linking

**align** center

En el simple ejemplo anterior, el conmutador compartirá todo el tráfico en todos los puertos y cada nodo lo recibirá en su enlace DtD. Si desea particionar aún más el tráfico, puede configurar VLAN adicionales en el conmutador para aislar el tráfico del puerto para que solo los nodos que deberían recibir tráfico específico lo vean. Por ejemplo, puede tener un sistema de videovigilancia (5) o un: abbr: *VoIP (Voz sobre IP)* Sistema PBX (6), y el tráfico de esos dispositivos solo debe pasarse a un conjunto específico de enlaces como se muestra en El diagrama a continuación. Las VLAN basadas en puertos asegurarán que el tráfico sea controlado y enrutado, en lugar de transmitirse a través de cada enlace.



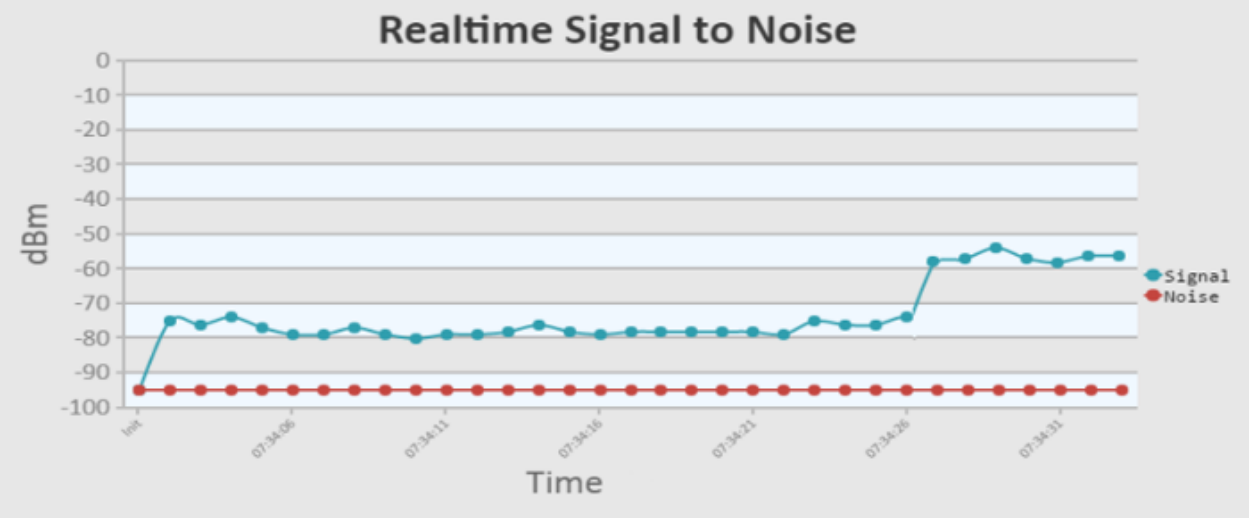

### 12.3.2 Polarización de la antena

La mayoría de las últimas AREDN | trade | dispositivos usan antenas de polaridad dual y: abbr: características *MIMO (entrada múltiple - salida múltiple)* en las radios que explotan la propagación por trayectos múltiples. Sin embargo, si está utilizando antenas de polaridad única con radios de "cadena simple", otra forma de lograr la separación de la señal para dispositivos colocados es orientar las antenas del sitio para que una esté polarizada verticalmente y la otra esté polarizada horizontalmente. Esto puede dar como resultado una separación de señal de hasta 20 dB. Debido al predominio de la polarización vertical en dispositivos WiFi comerciales, AREDN | trade | nodos pueden lograr un rendimiento ligeramente mejor utilizando polarización horizontal con una línea de visión clara. Puede probar ambas polarizaciones para ver cuál produce un mejor rendimiento frente al ruido artificial en su entorno específico. Tenga en cuenta que las antenas en ambos lados de un enlace de radio deben estar orientadas de la misma manera.

### 12.3.3 Alinear nodos vinculados

El AREDN | trade | web interfaz proporciona información útil cuando se alinean dos nodos que se están instalando para formar un enlace. En la página ** Estado de nodo **, haga clic en el botón ** Gráficos ** para ver el gráfico * Señal en tiempo real a ruido *. Lentamente gire e incline su antena, haciendo una pausa para ver las métricas de la señal. Una vez que vea la mejor señal, como se muestra a continuación, puede bloquear la antena en su posición. Si desea enfocarse en la

posición de la antena sin tener que mirar el gráfico SNR, también puede habilitar la función * SNR Sound * y alinear la antena con el tono de tono más alto. Dependiendo de la implementación, una relación señal / ruido de 15 dB es adecuada para pasar datos a velocidades en el rango de 5 a 20: abbr: *Mbps (Megabits por segundo)*.




## 12.4 Consejos de planificación de canales

Si hay dos torres o áreas de cobertura celular dentro del alcance, configure los nodos con diferentes canales para evitar un bajo rendimiento.

Según el propósito de su red, intente crear rutas confiables a las ubicaciones donde se necesitan datos. Utilice la separación de canales y el enlace DtD de los nodos colocados para evitar la contención del canal de RF. Las bandas de 3,4 GHz y 5,8 GHz proporcionan los canales más compartidos para su uso en AREDN | trade | redes.

- Si necesita una amplia cobertura local para un área muy grande, puede instalar antenas sectoriales en una torre en un sitio alto: por ejemplo, tres paneles con un ancho de haz de 120 grados cada uno. Un enlace DtD vincula los sectores en la torre y se utilizarían diferentes canales para cada sector. De esta manera se evitaria la congestión en los canales.
- Considere poner cada área de cobertura local en su propio canal para minimizar la interacción entre zonas.
- Si está instalando enlaces punto a punto de larga distancia para conectar islas de malla, asegúrese de usar una banda o canal separado para el enlace troncal. Este tipo de enlace tiene un único propósito: transportar la mayor cantidad de datos lo más rápido posible de un extremo al otro. Elimine cualquier tipo de contención de canal para que estos enlaces puedan lograr un alto rendimiento.
- Recuerde que una ruta de múltiples saltos a través de la red debe tener una buena calidad de señal en cada tramo del viaje. No puede esperar un rendimiento adecuado a través de una

serie de enlaces de baja calidad. Por ejemplo, si atraviesa tres enlaces que tienen: métricas abbr: *LQ (calidad de enlace)* 65%, 45% y 58%, su agregado: abbr: *LQ (calidad de enlace)* será 17%, lo que no se puede usar . Idealmente, el agregado: abbr: *LQ (calidad de enlace)* debe tener al menos un 80% para tener un enlace que admita las aplicaciones y servicios que necesita.

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

## 13.1 Creating a Path Profile

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

- Latitude and Longitude
- Antenna AGL (height Above Ground Level)
- Frequency
- Transmit Power
- Line Loss

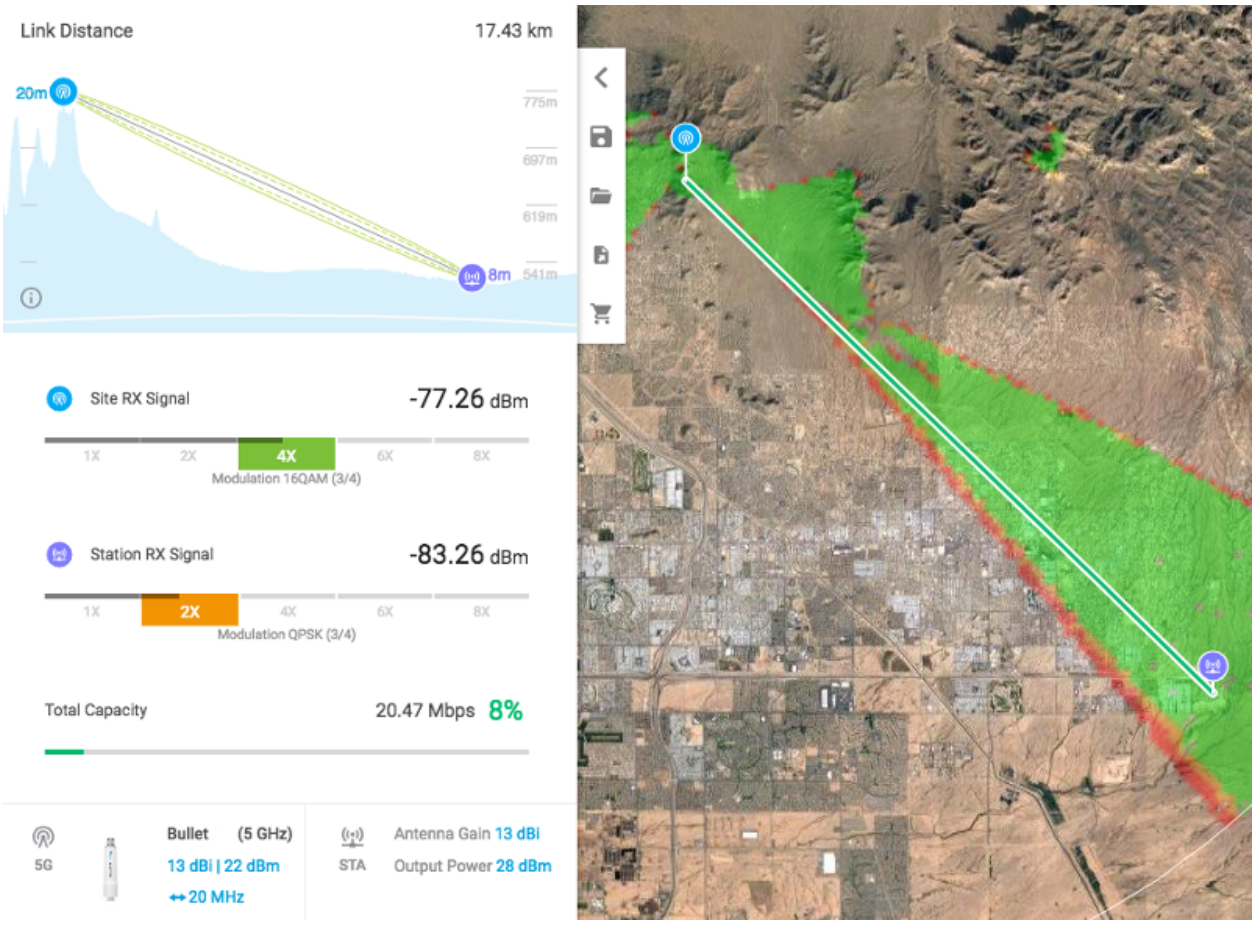
- Antenna Gain
- Receiver Sensitivity

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

### 13.1.1 Ubiquiti AirLink Tool

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

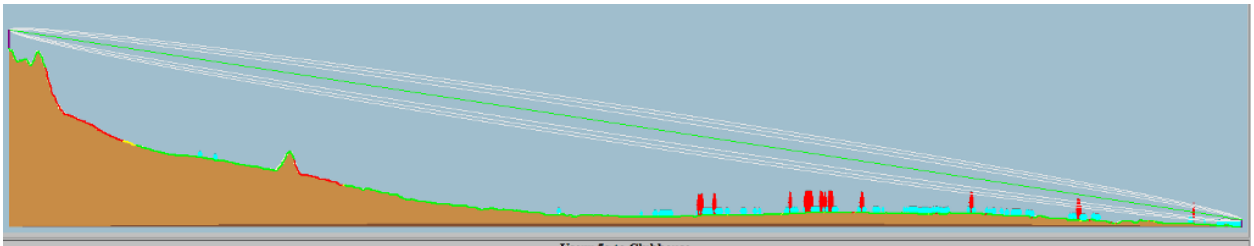
The path profile is color coded to indicate whether the link quality is adequate. It displays the link distance, line of sight, as well as the Fresnel Zone and 60% clearance area. It also estimates the signal levels at each endpoint and the predicted throughput for the link. An example *AirLink* path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



### 13.1.2 VE2DBE's Radio Mobile Tool

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

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



**Usery 5g to Clubhouse**

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

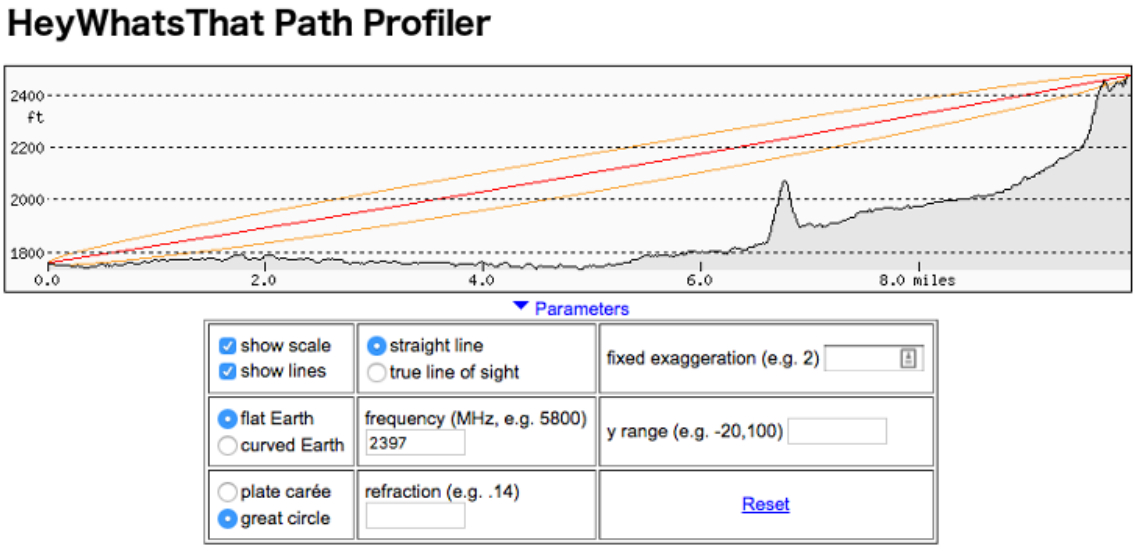
| **Radio system** | | **Propagation** | |
|---|---|---|---|
| TX power | 28.45 dBm | Free space loss | 132.52 dB |
| TX line loss | 0.50 dB | Obstuction loss | -3.16 dB |
| TX antenna gain | 20.00 dBi | Forest loss | 0.00 dB |
| RX antenna gain | 16.00 dBi | Urban loss | 1.00 dB |
| RX line loss | 0.50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

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

### 13.1.3 HeyWhatsThat Path Profiler

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

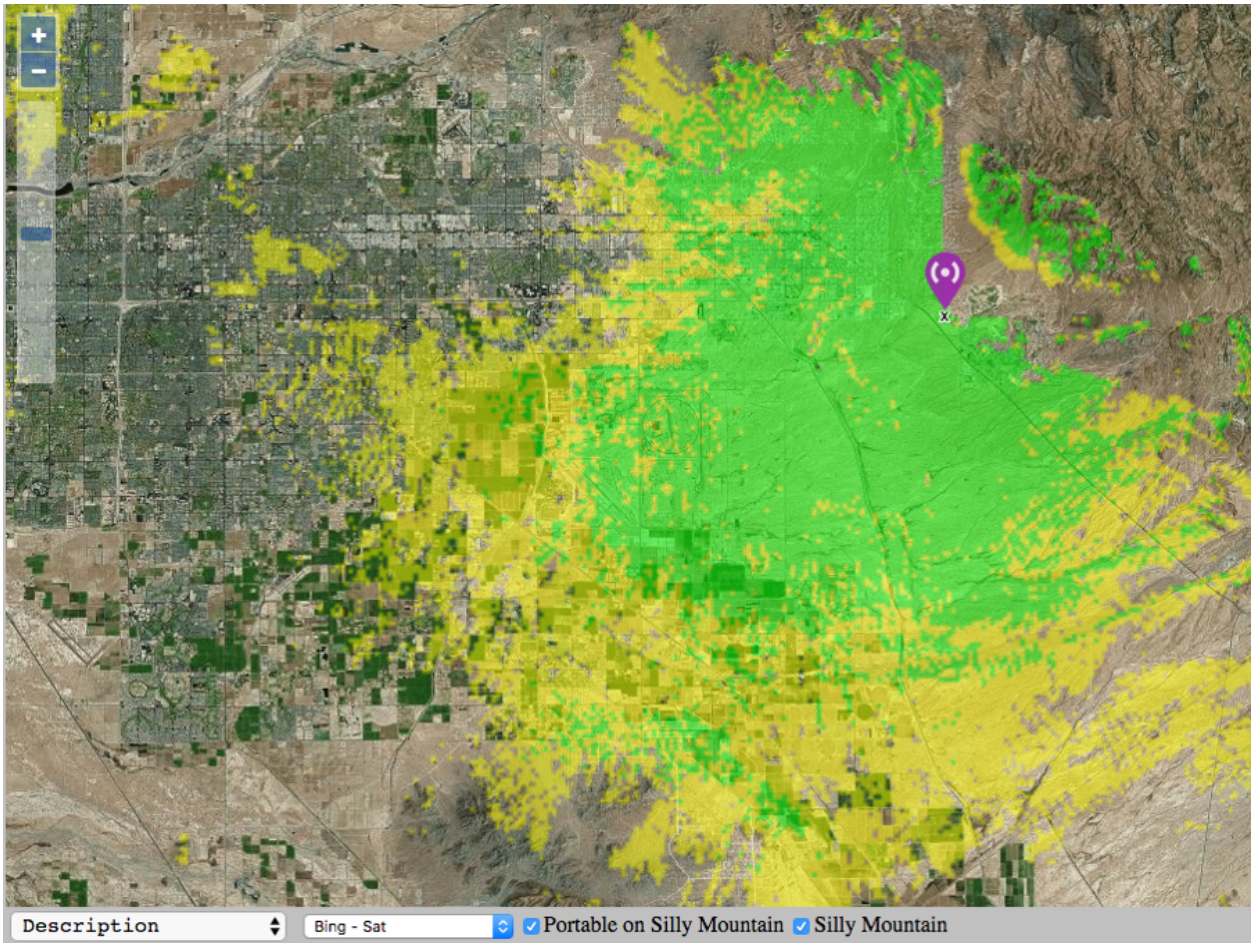
Simply click on the map at the bottom of the webpage to add an endpoint for each side of your link. Once an endpoint has been added, it can be moved by clicking and holding the endpoint while dragging it to the new location on the map. After adding your endpoints you will see the path profile displayed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




## 13.2 Determining Node or Network Coverage

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

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


Description
Bing - Sat
Portable on Silly Mountain
Silly Mountain

# CHAPTER 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 (Voice over IP) 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.

**Note:** The consideration above is especially important if you are deploying a service which regularly queries other nodes across the network. For example, if you deploy a network management system which polls metrics from remote mesh nodes, you need to carefully consider how many metrics you poll and how often you request them. Realize that polling dozens of metrics from each

node every few seconds is likely to degrade mesh performance. Be sure to let node owners know what you are planning to do and get their permission/agreement for your polling schedule.

---

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

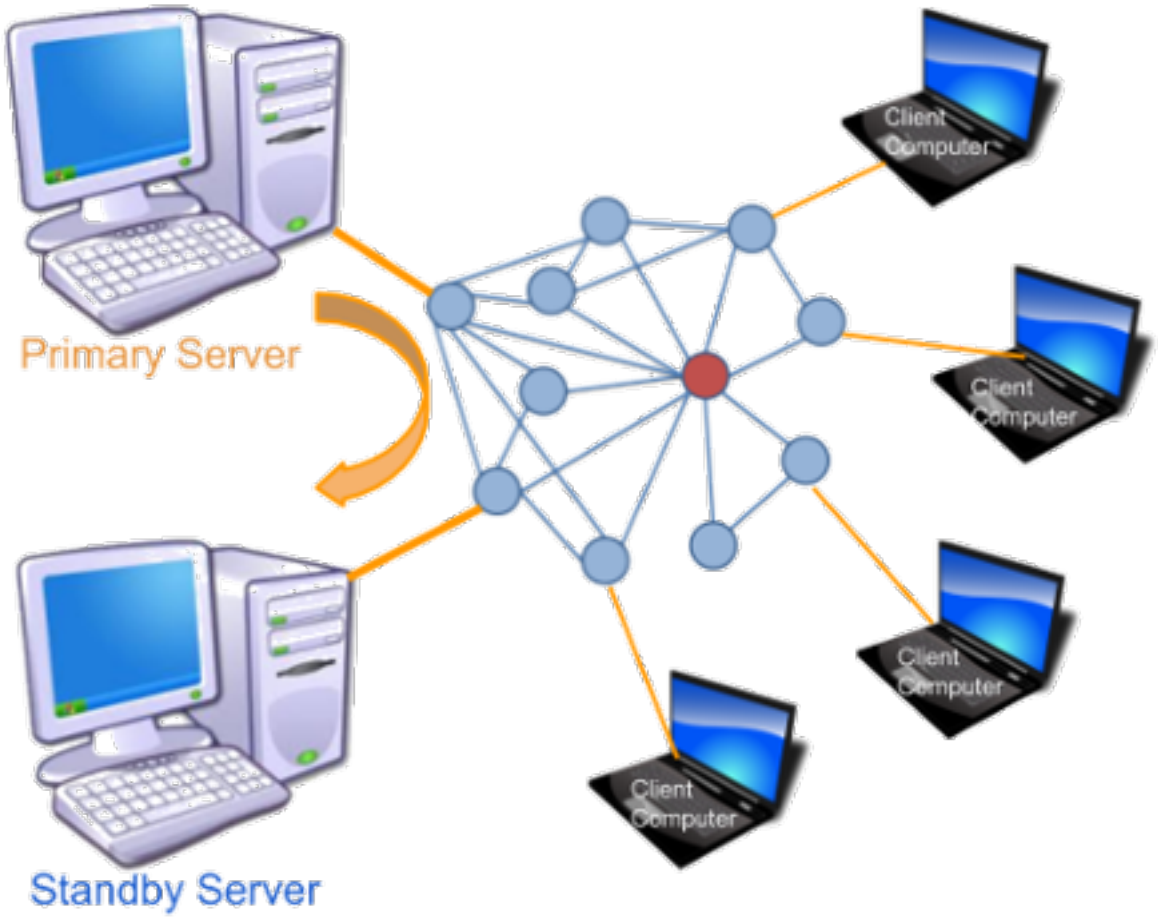
**Keeping Multiple Servers in Sync**

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

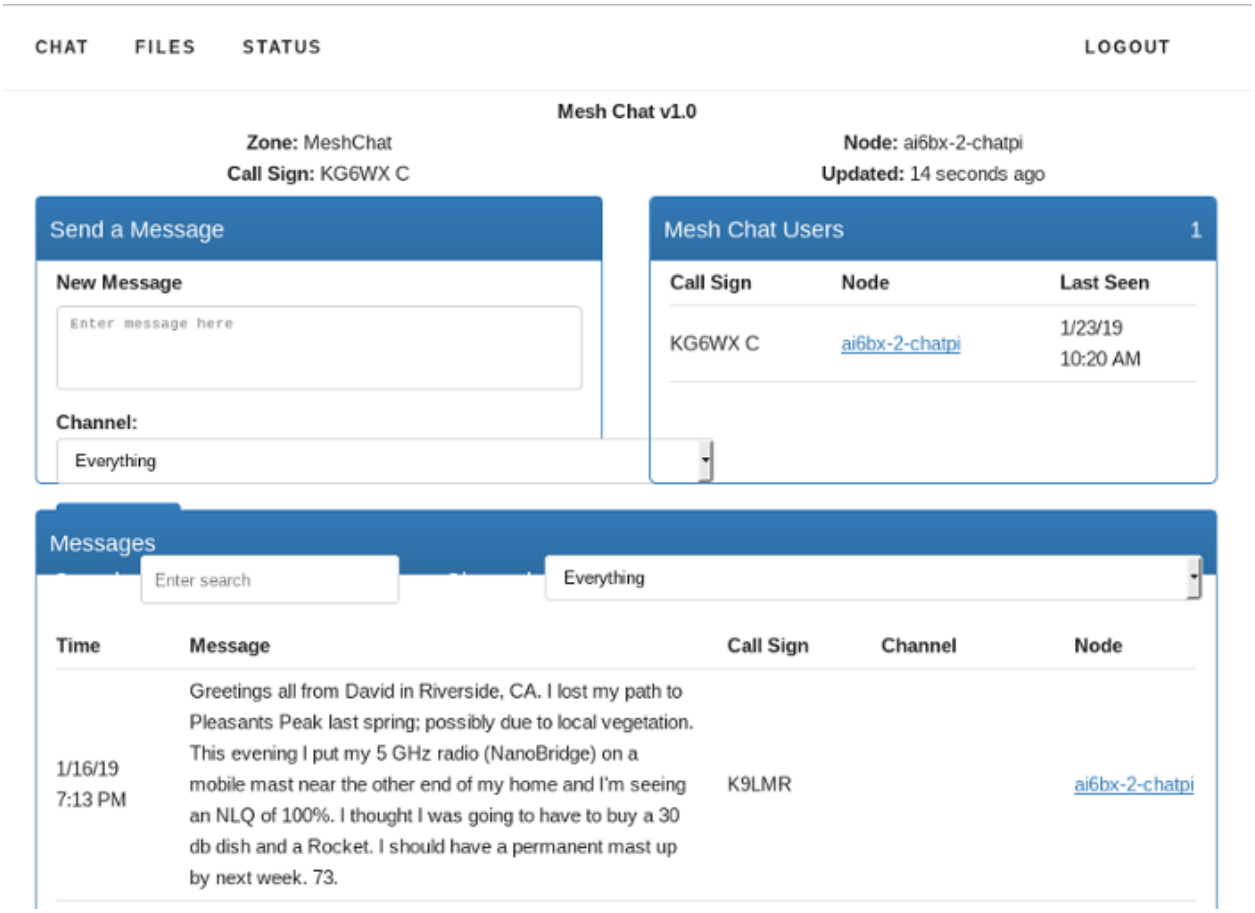
MeshChat has become the primary chat service for AREDN® networks because it was written specifically for mesh communication. Users access MeshChat via web browser, and the service runs on the mesh node itself or on a LAN-connected Raspberry Pi computer. After logging in by entering a call sign, send 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

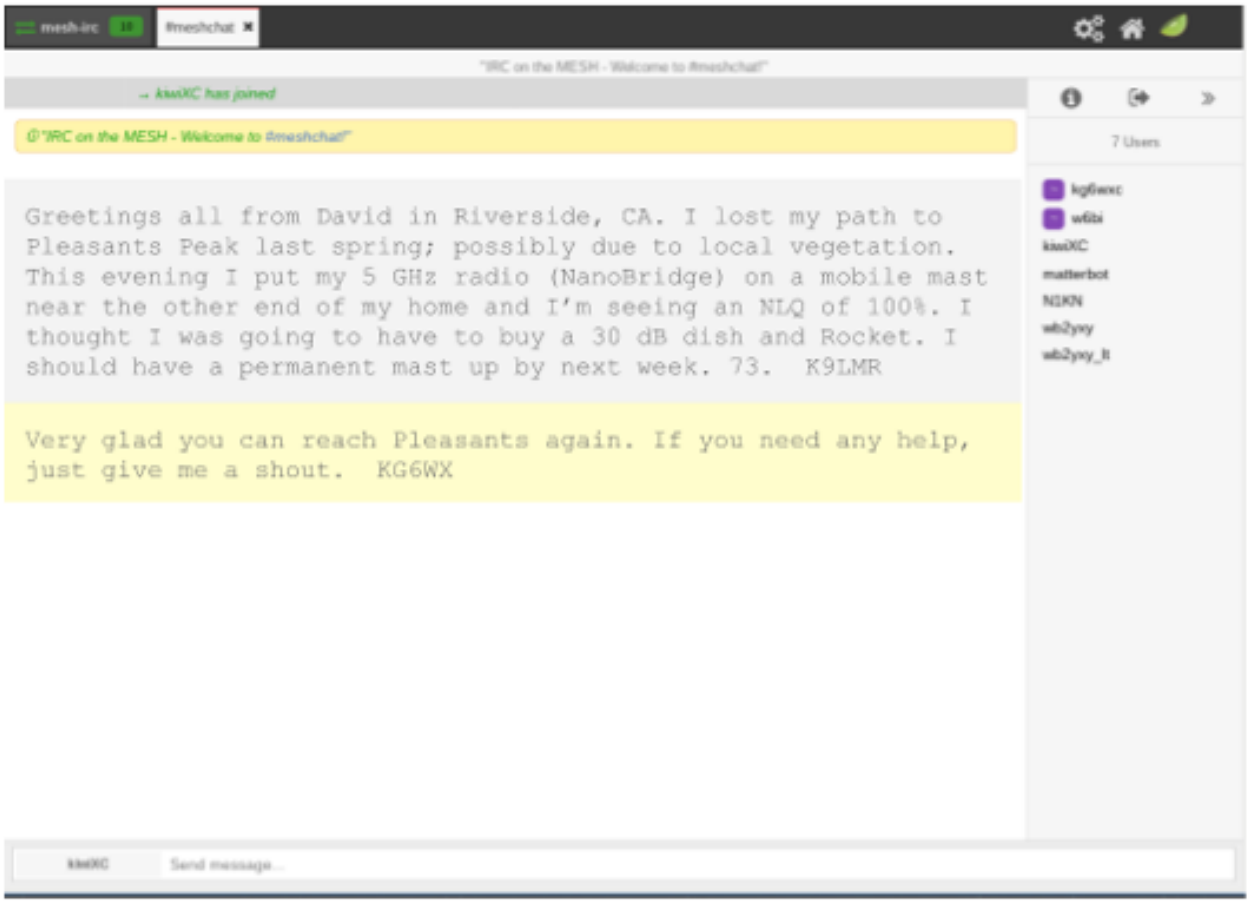



## 15.2 Internet Relay Chat

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.

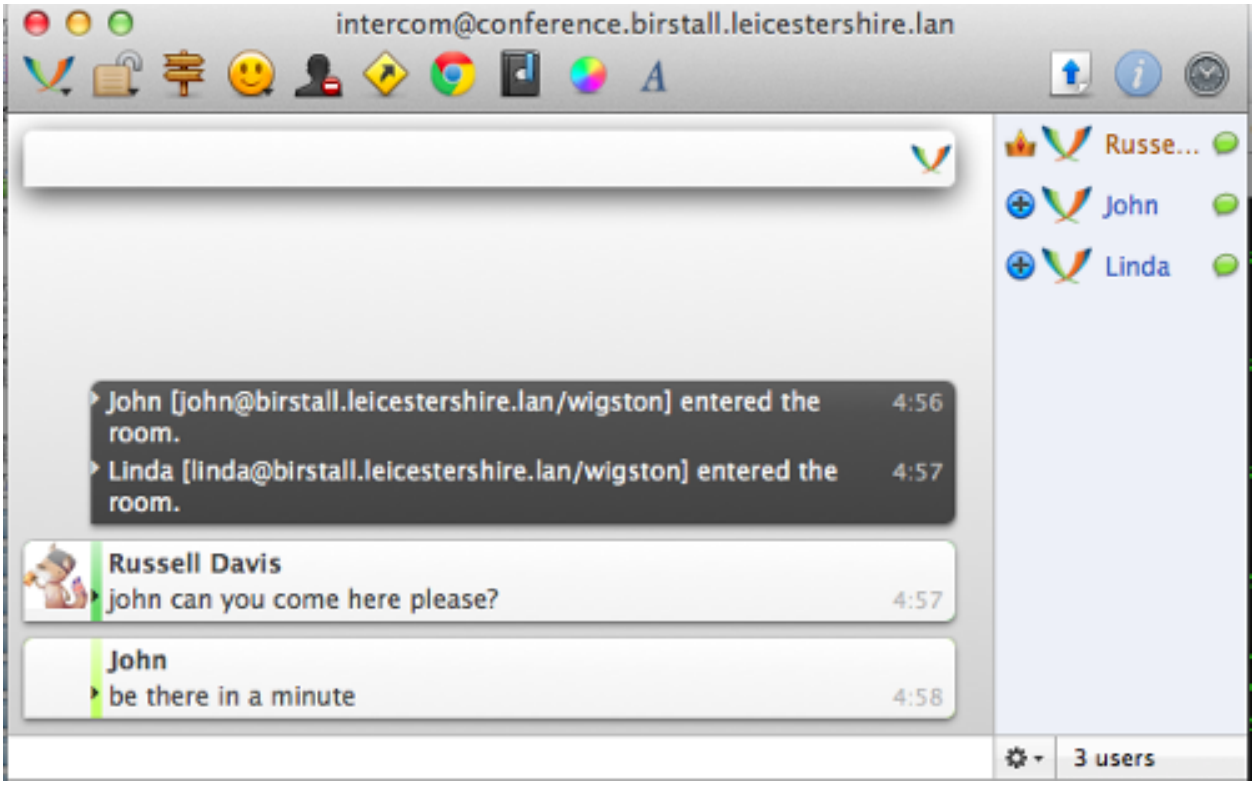
A wide variety of features and functions are available with these and similar chat programs, including various zones, channel types, and user roles. For additional information about IRC services, visit these links: IRC Servers and IRC Clients




## 15.3 Jabber/XMPP

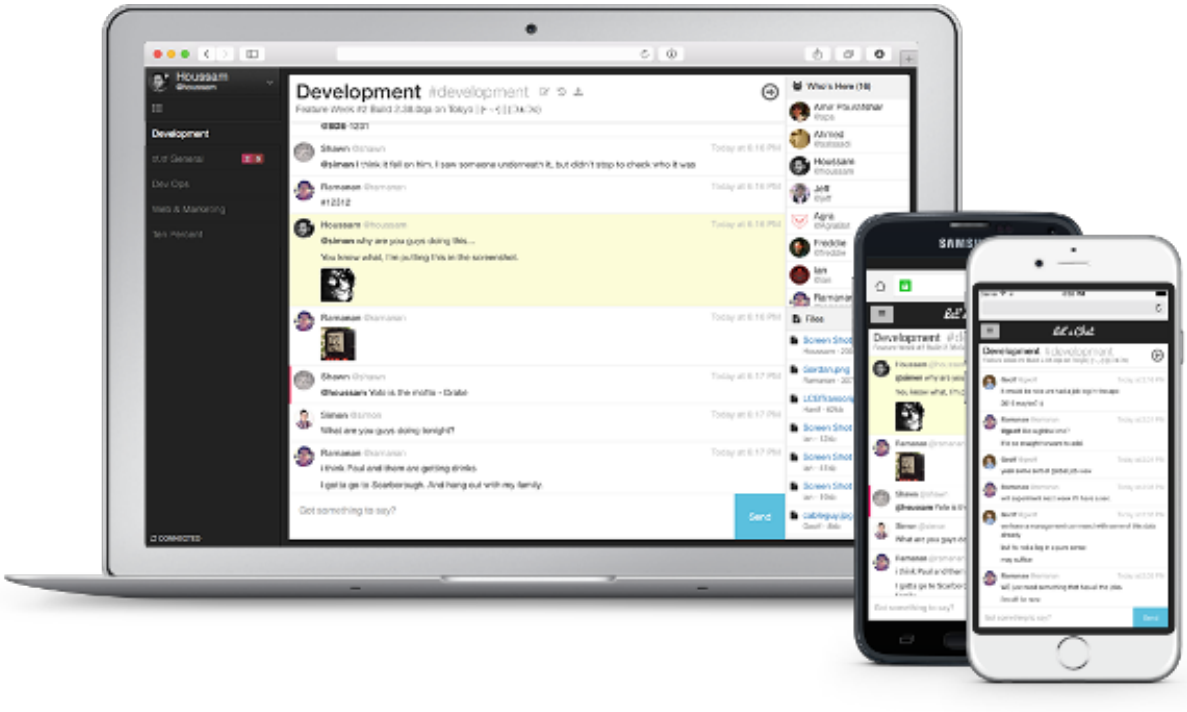
Originally known as Jabber, XMPP servers have been around for a long time but are fully compliant with modern messaging standards thanks to a large community of developers worldwide. These servers provide one-to-one messaging as well as group chat sessions. User lists have activity and presence indicators,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

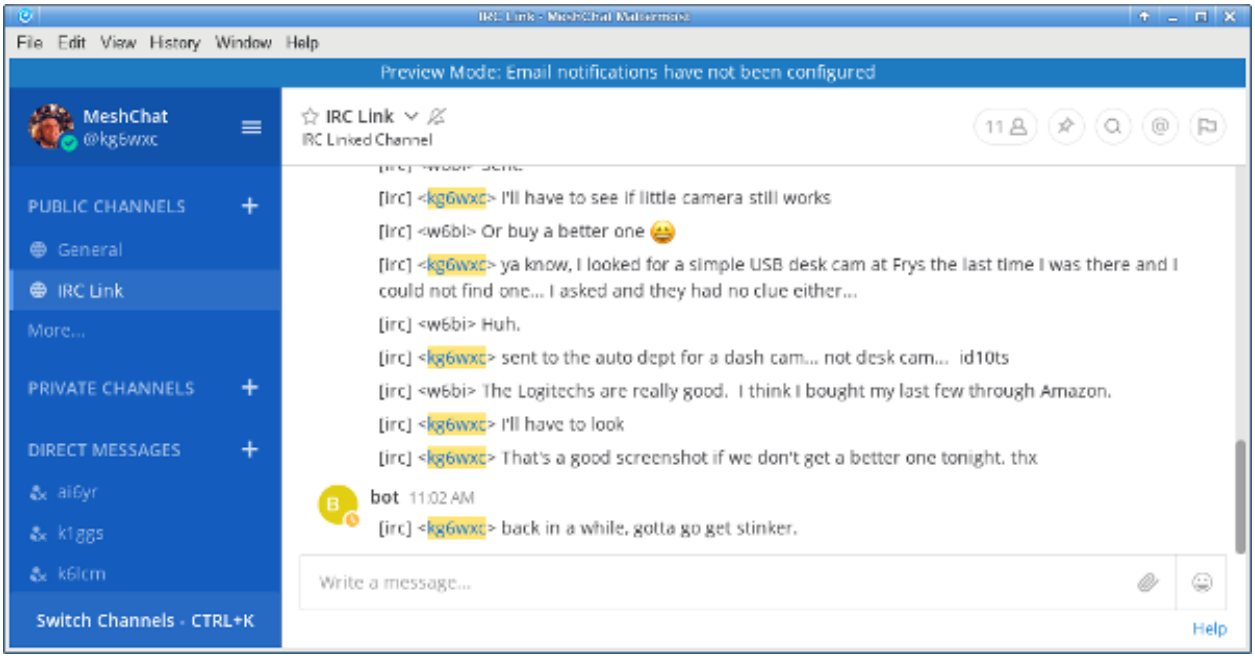
*Let's Chat* is an open source messaging service for small teams. It provides one-to-one communication between XMPP users as well as group messaging and @mentions in a variety of chat rooms. Searchable conversation history is available, in addition to text and image pasting, user activity notifications,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

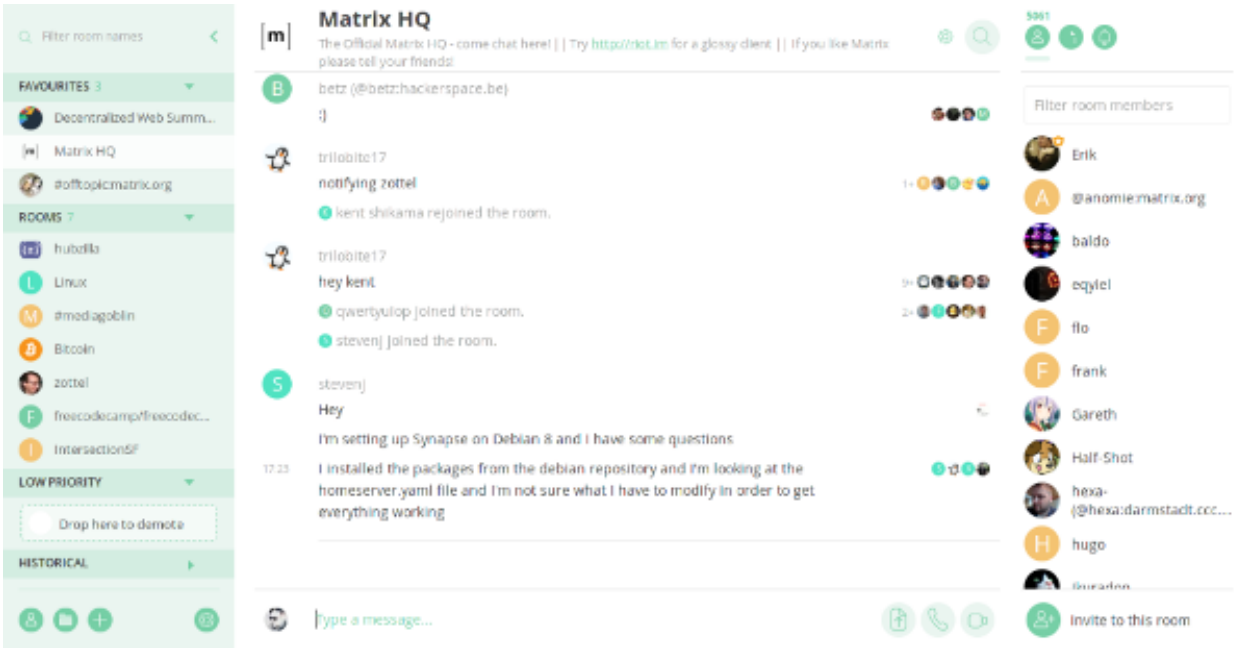
Mattermost supports @mentions, and channels are available for organizing conversations which can be topic-based, group-based, or event-based. Notifications indicate user presence and activity. File sharing is provided for PDF and text files, as well as audio, video, and image files. For additional information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

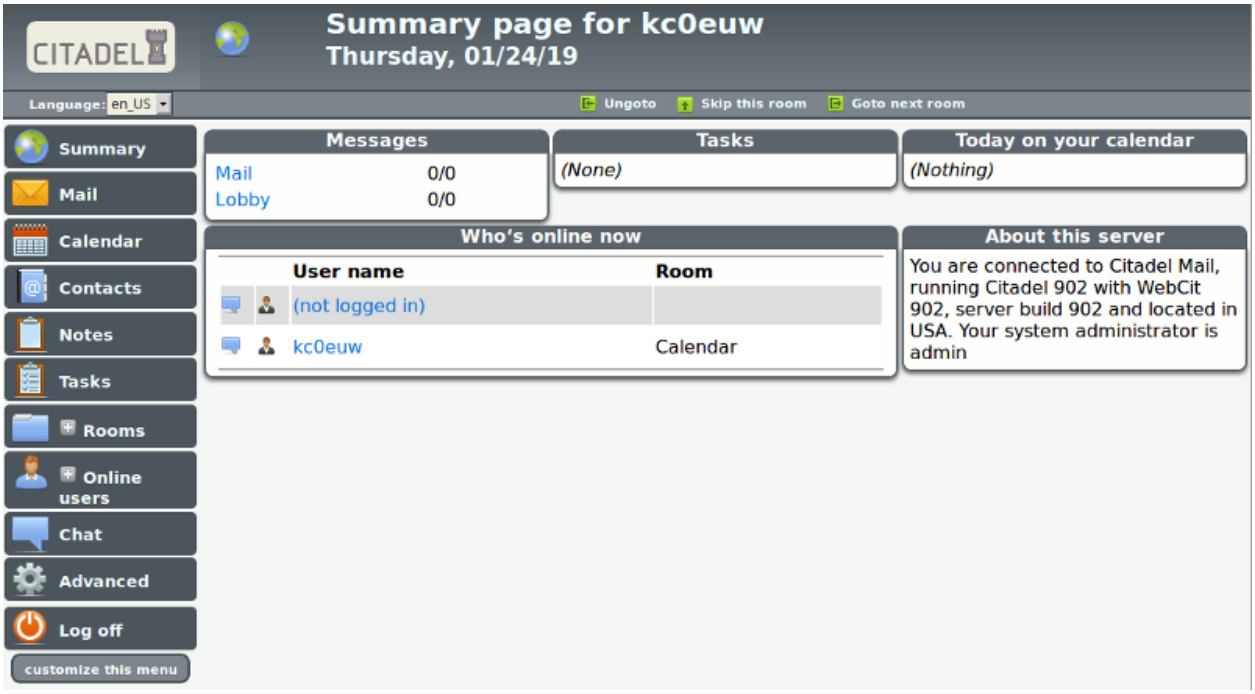
Email operates on a client-server model. Users create or read their messages on some type of client program, although this software could be hosted on a network web server and accessed through a user's web browser rather than requiring a standalone email program to be installed on the client computer. 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.

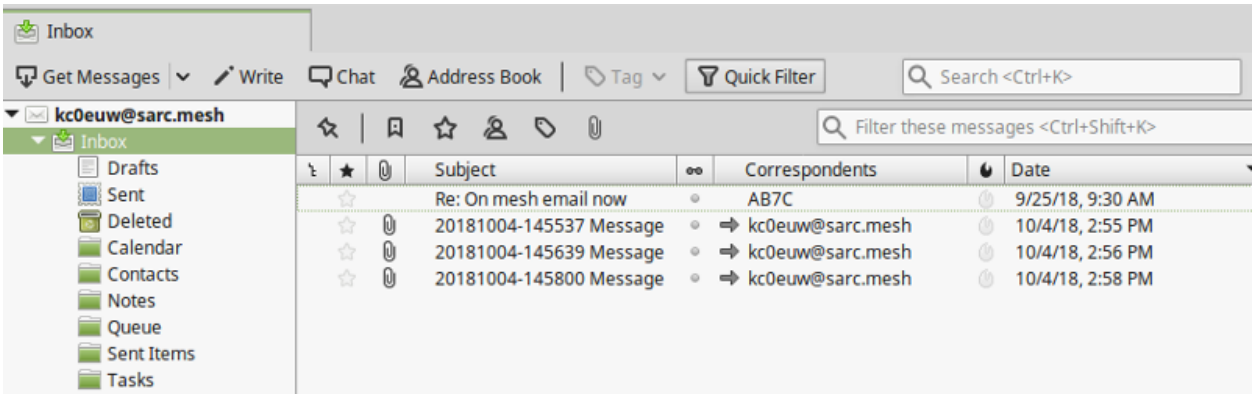
**SMTP** In order to implement an email server you will need to select a software package to handle the Simple Mail Transfer Protocol. You can select one of the example open source packages from the list below, or you can implement another SMTP agent of your choice.

- 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.

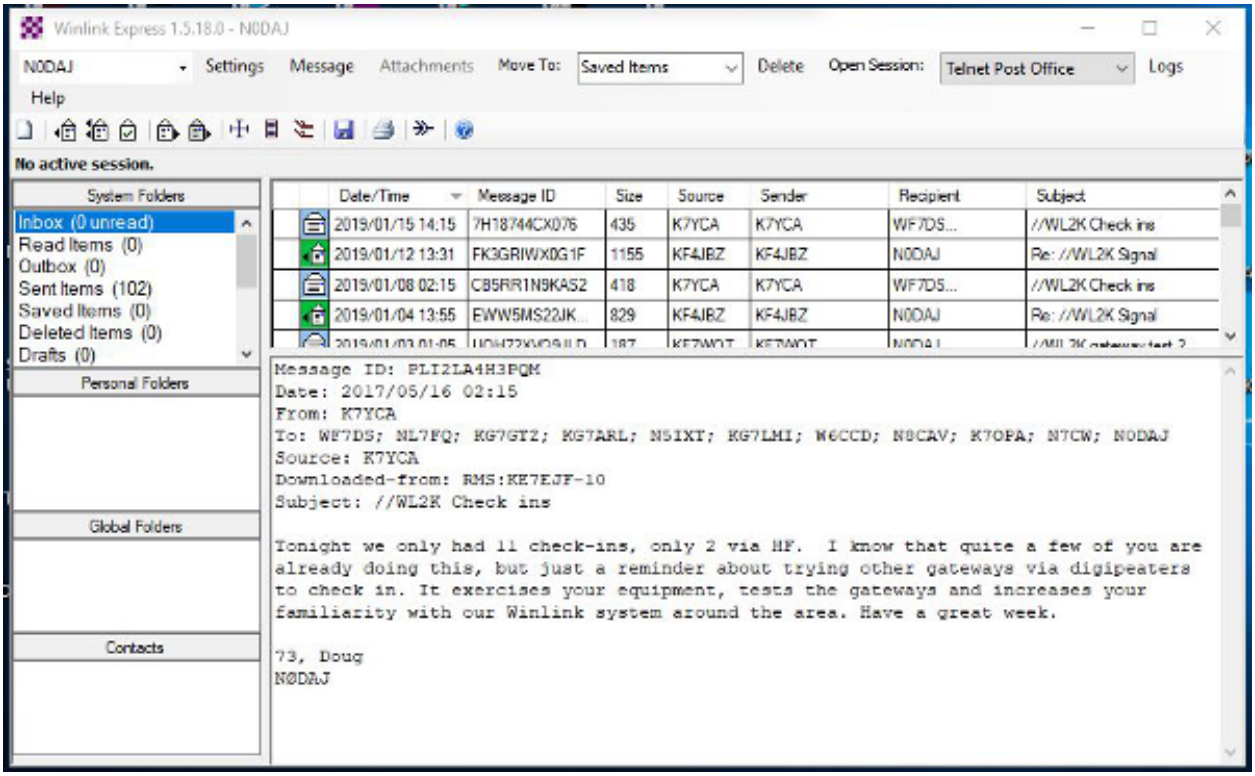
You will need to have detailed knowledge and skills when building your own open source email server, with the advantage of having complete control over everything on the system. There is some administrative overhead for creating and maintaining all user email accounts as well as handling other management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

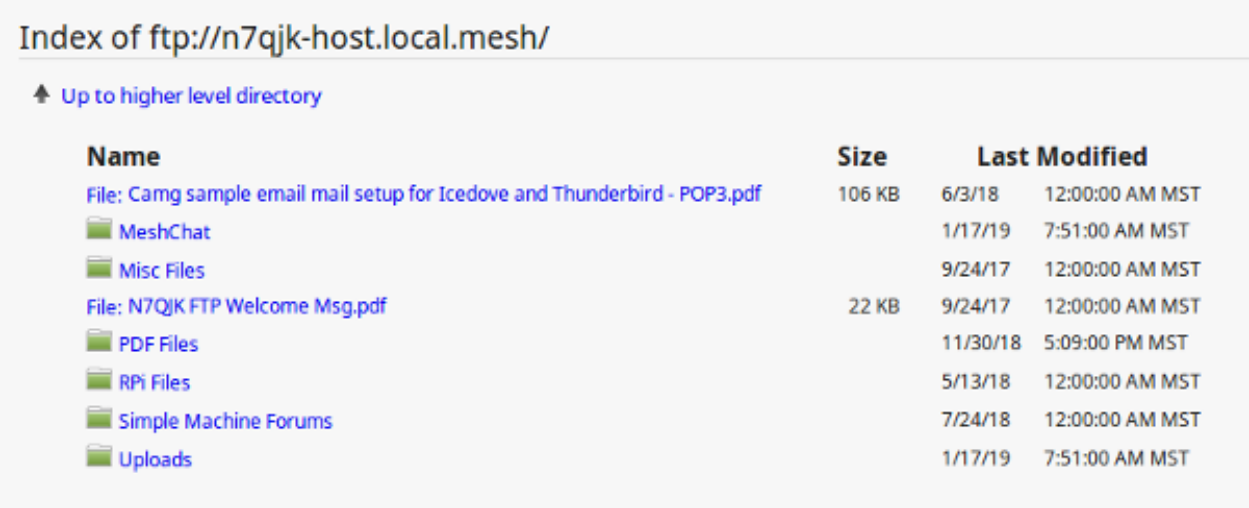
File transfer protocols themselves have minimal impact to network performance, but downloading a very large file across a mesh network could have a major performance impact. Transferring text files, and especially compressed text, should have minimal impact to the network, but a network could experience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

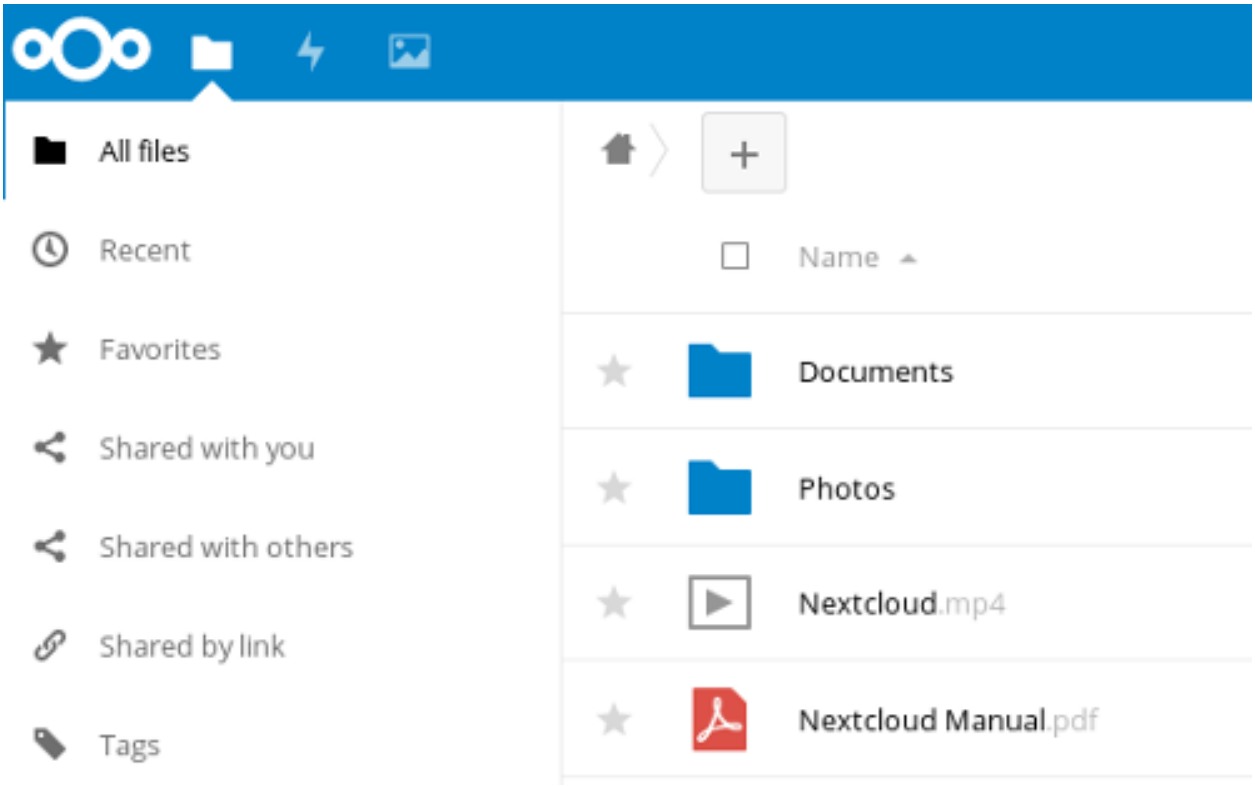



## 17.2 Web Services

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.

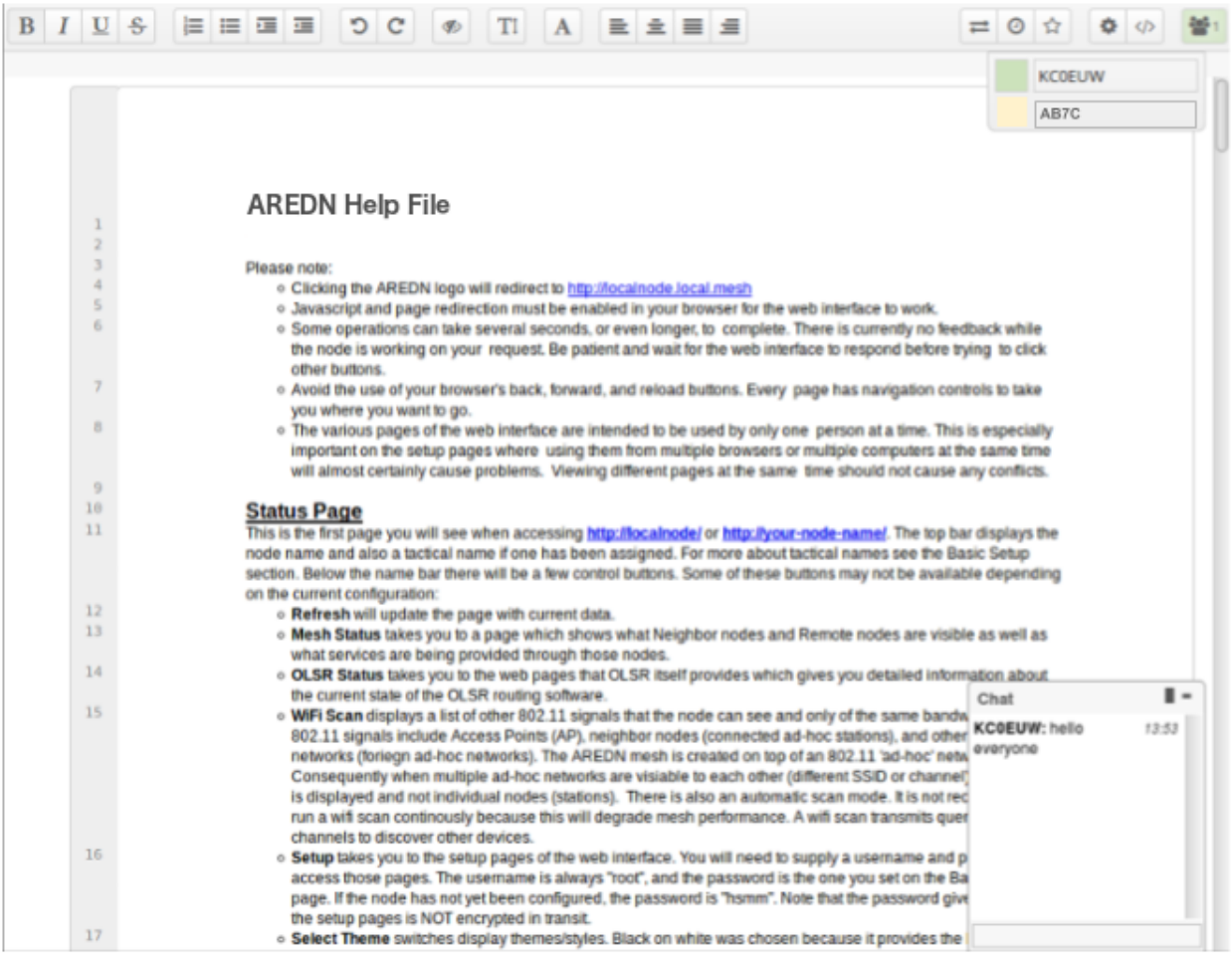



## 17.3 Collaborative Computing

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.


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

# CHAPTER 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 network. The phrase Voice over IP (VoIP) encompasses a collection of technologies capable of encoding and delivering realtime multimedia content across a digital network. When you have an established need for this type of communication, and if your mesh network is capable of supporting it, there are many reliable options for implementing VoIP and video conferencing.

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

## 18.1 VoIP Server

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

Dozens of full-length books have been written about Asterisk, so it is widely documented. It also serves as the underlying communication engine for several other software PBX 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.

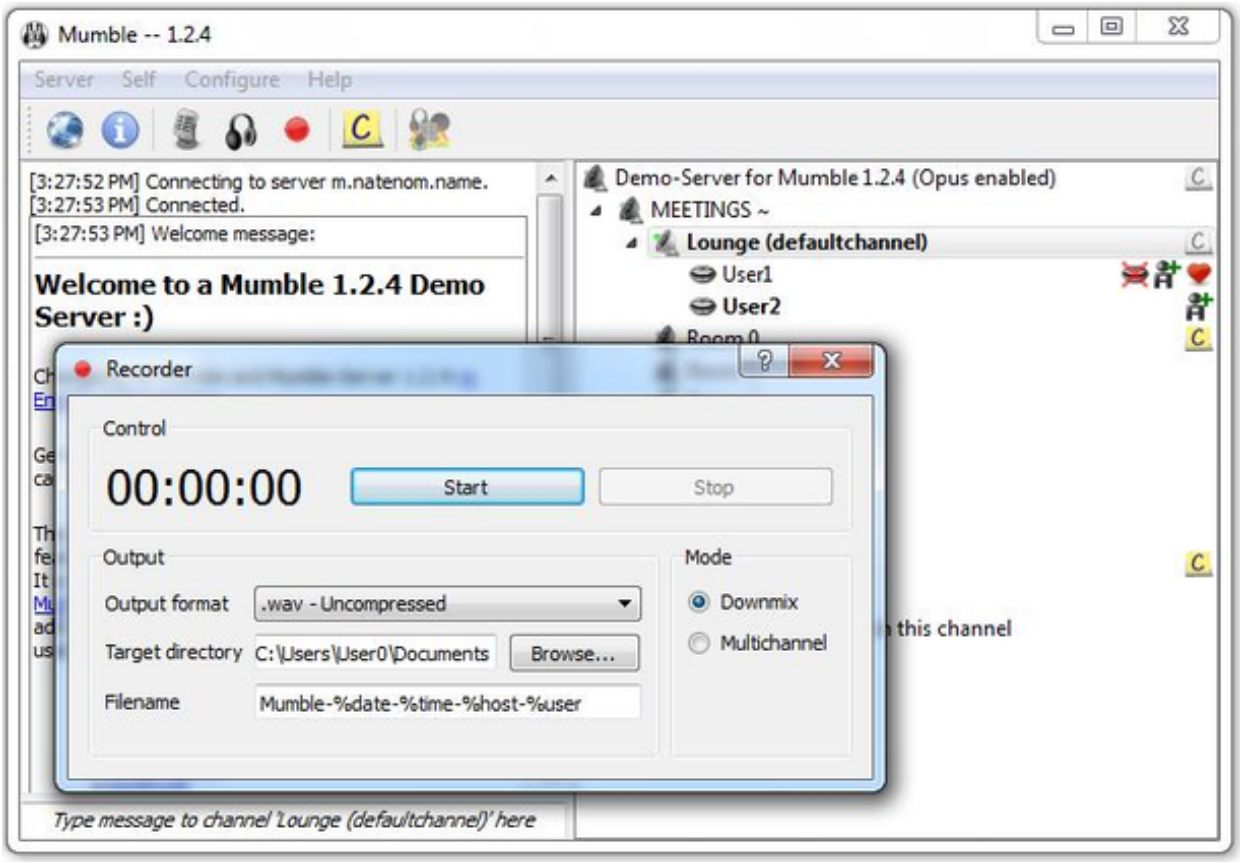

## 18.2 VoIP Endpoints

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.

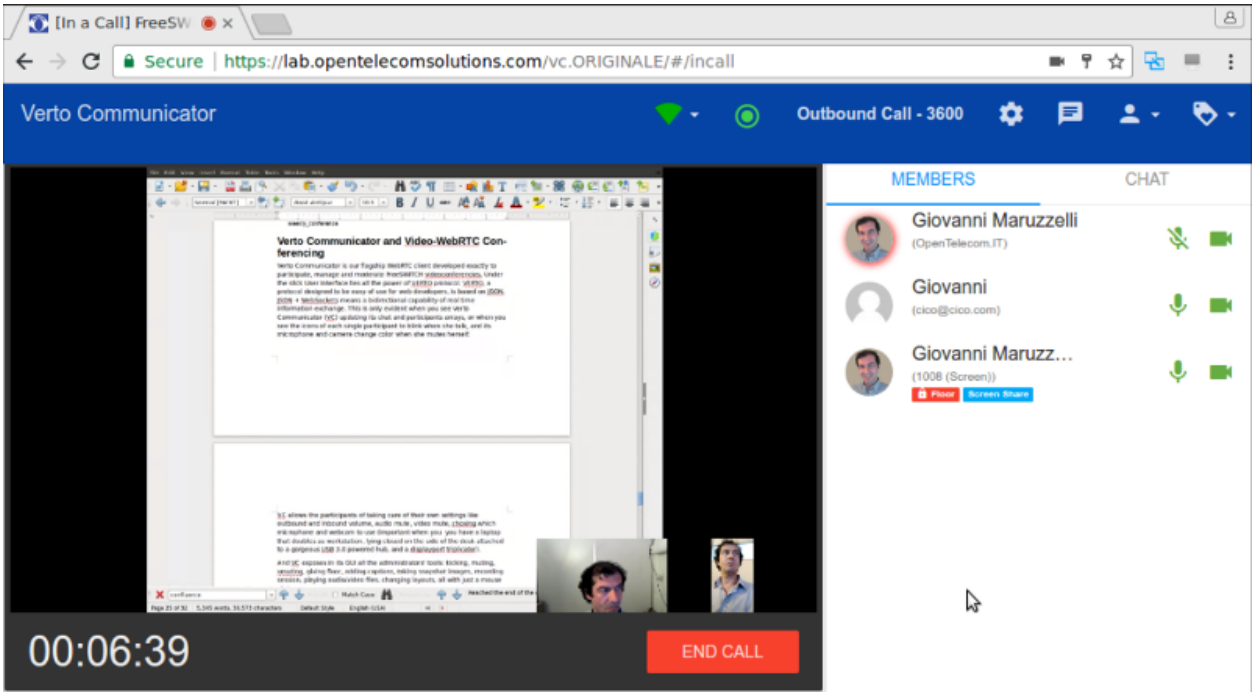
Hosting Mumble locally requires downloading the *Murmur* server, which is included as an option in the Mumble installer. The primary users of Mumble are Internet video gamers who want to communicate with each other during game play. However, it can also be used as a non-gaming voice communication service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




## 18.3 Video Conferencing Software

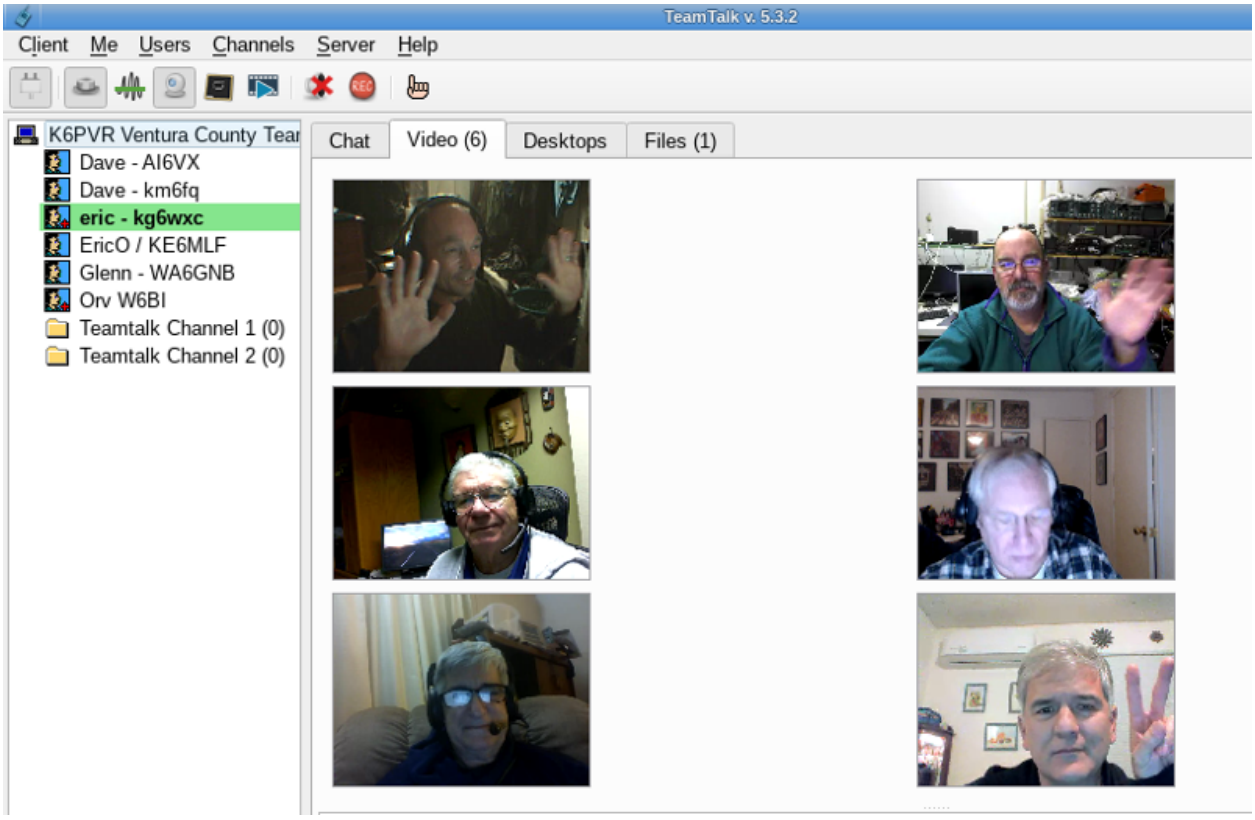
**FreeSWITCH Server** FreeSWITCH is a recent communication platform that can be used to build voice PBX systems with voice response menus, video conferencing with chat messaging and screen sharing capabilities, and full WebRTC support. Its modular design makes it possible to install only what is required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

## 19.2 Video Display Software

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

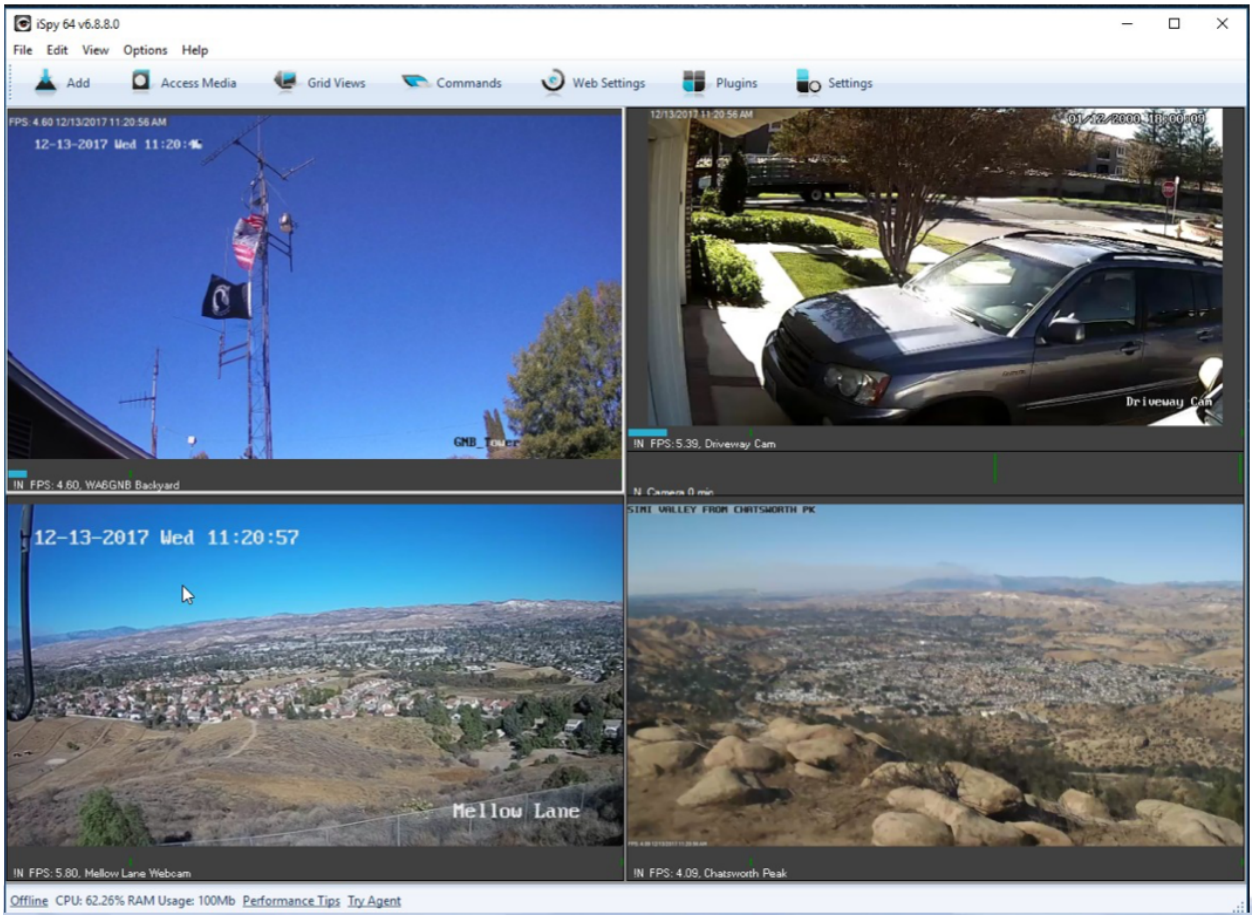
### 19.2.1 iSpy

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

The Windows program provides a "surface" or workspace where you add and configure multiple cameras or microphones. You can then monitor and interact with them to display live video or listen to live audio from network devices. Multimedia streams can be recorded locally for future use, and PTZ cameras can be manipulated with controls in the UI. Motion detection can also be configured, which provides a method for automatically recording multimedia snippets when specific events occur.

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

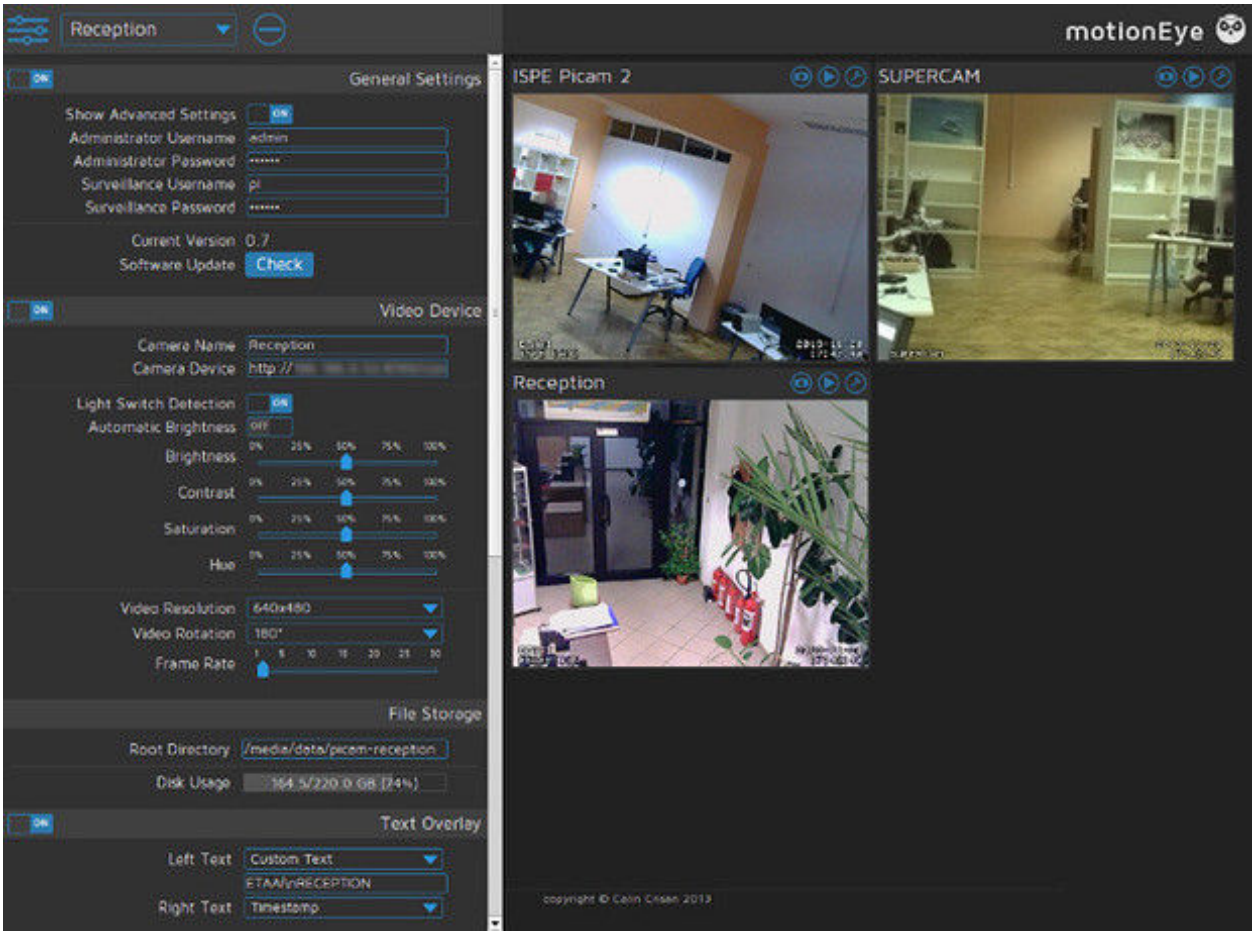
In the lower right video stream on the iSpy display below you can see the smoke plume from the 2017 Thomas Fire in California, which was recorded by a camera on the local AREDN® network. For additional information about iSpy, visit this link: iSpy.




### 19.2.2 MotionEye

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

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

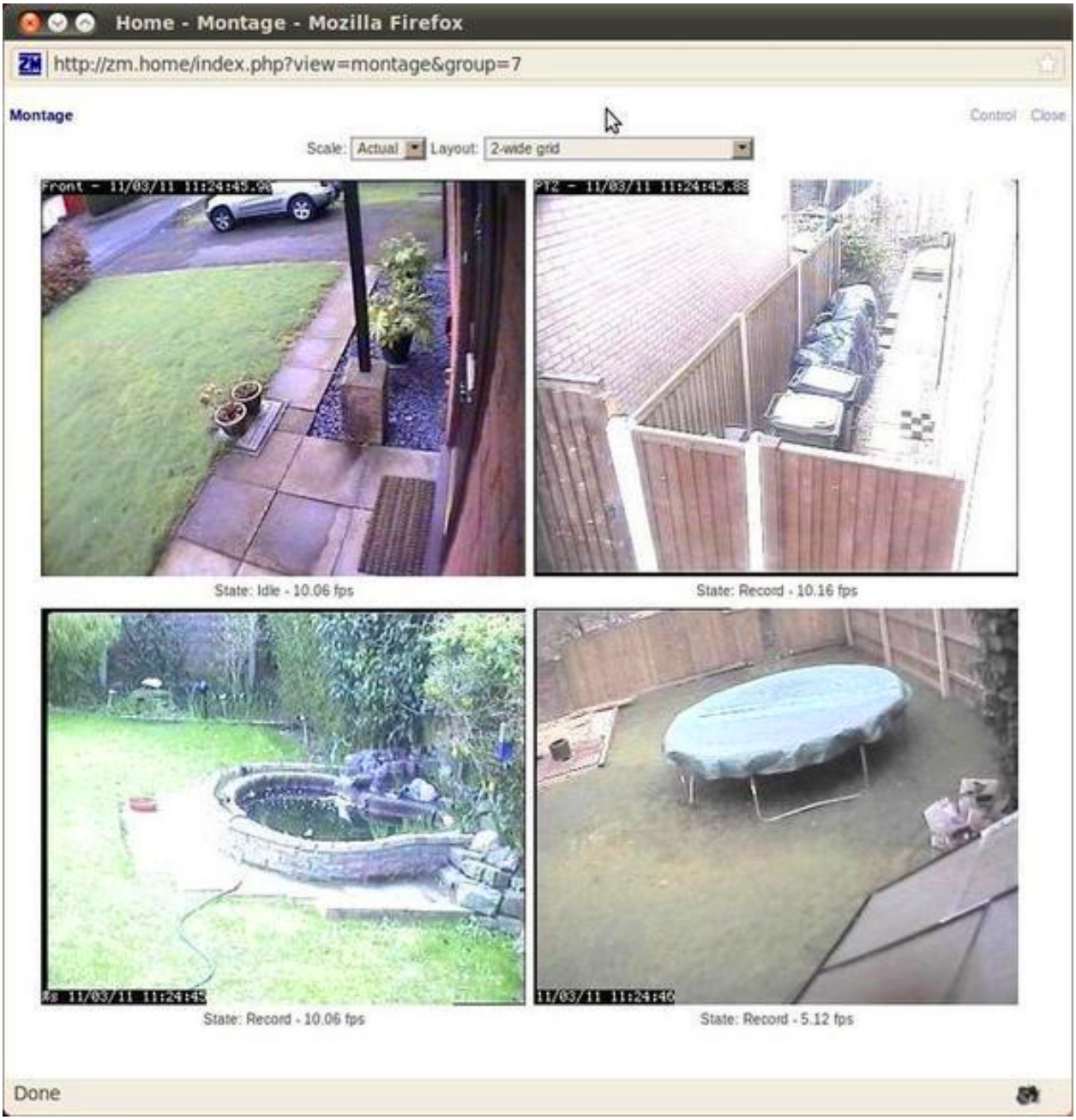



### 19.2.3 ZoneMinder

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

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

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



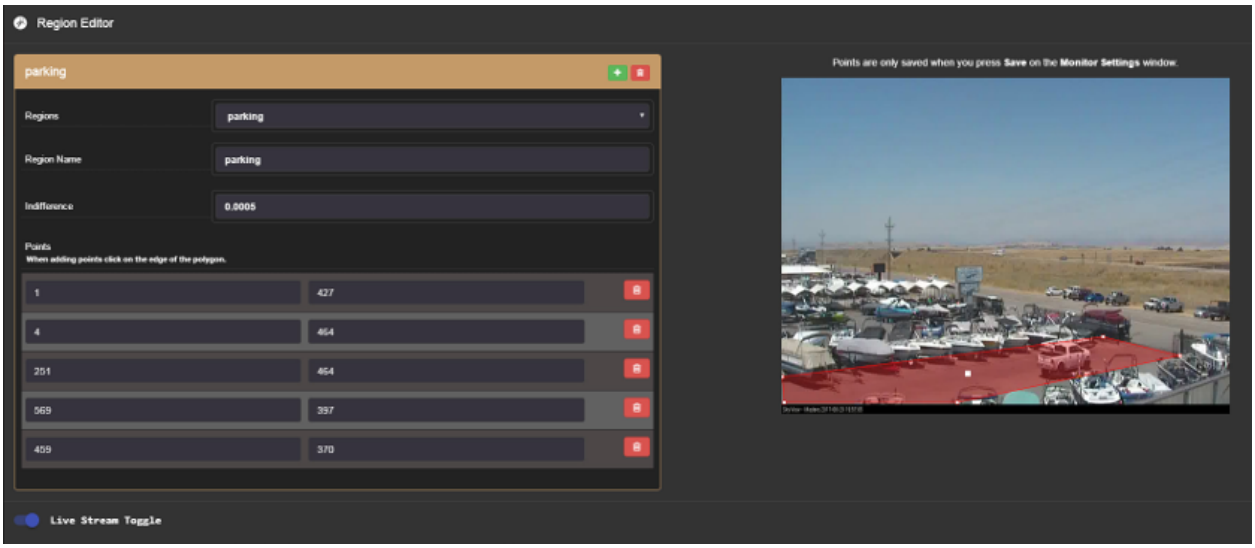

### 19.2.4 Shinobi

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 *NC* 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.

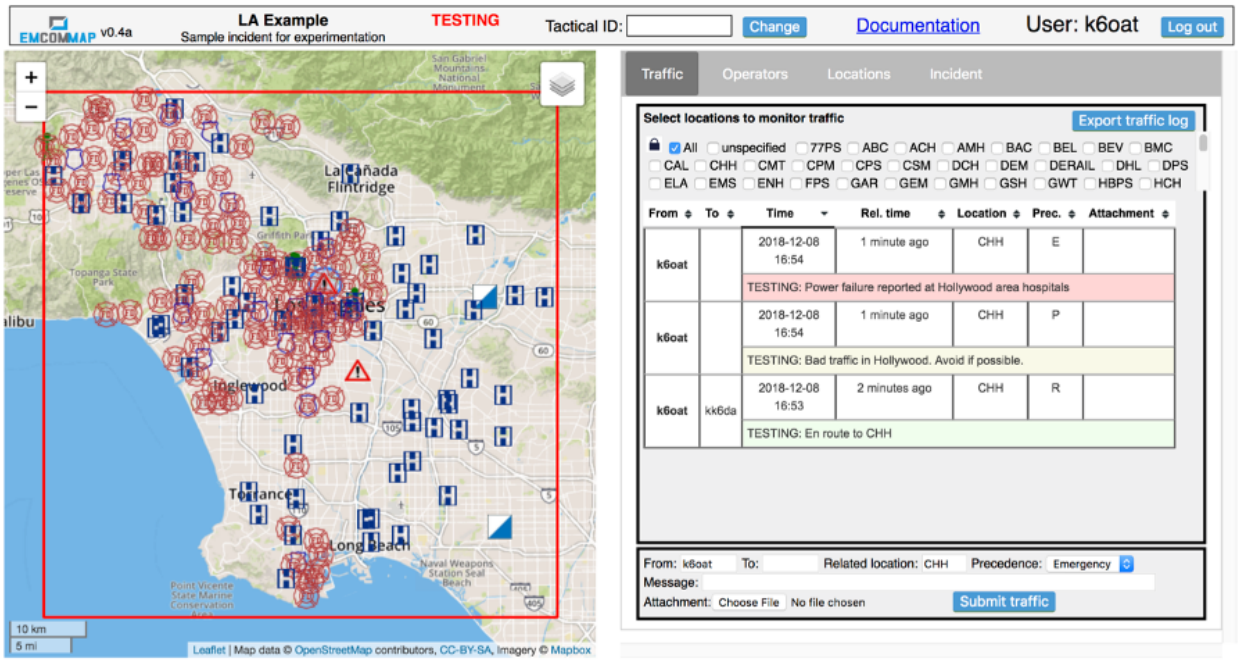
Served agencies have been using Computer Aided Dispatch (CAD) software for quite some time, and it has become their preferred method for managing events and incidents within their jurisdiction. In emergencies when electrical power or mission-critical facilities become unavailable and agencies are forced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




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

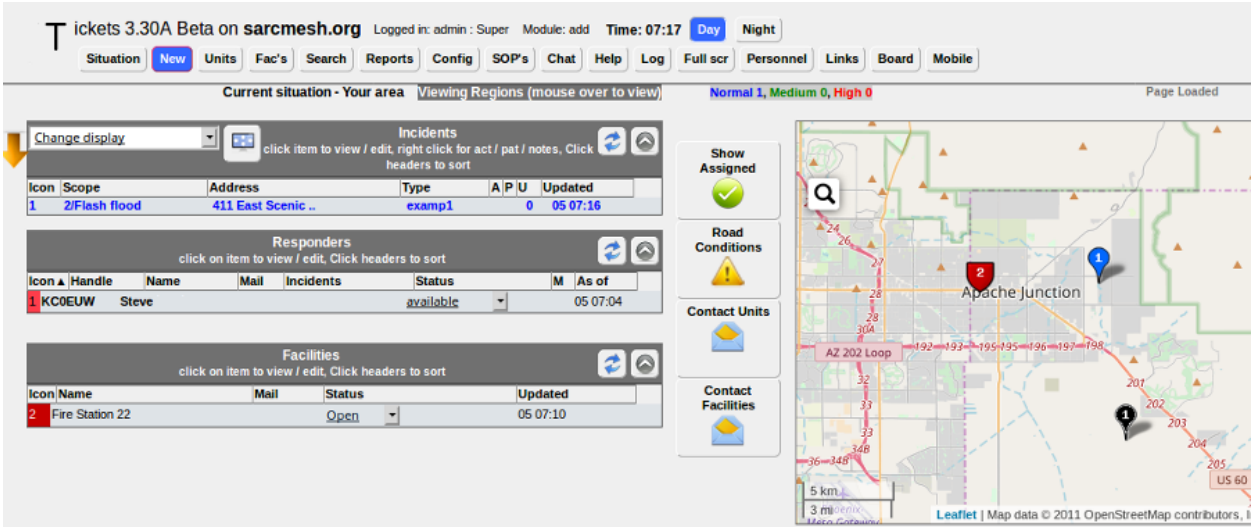
All communications are tracked and can be exported in spreadsheet format for offline use. Message traffic can be filtered to view specific messages for selected locations, and the traffic table can also be sorted for viewing the details based on information in any column. Message severity levels and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

## 20.2 Open ISES Tickets

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

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

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

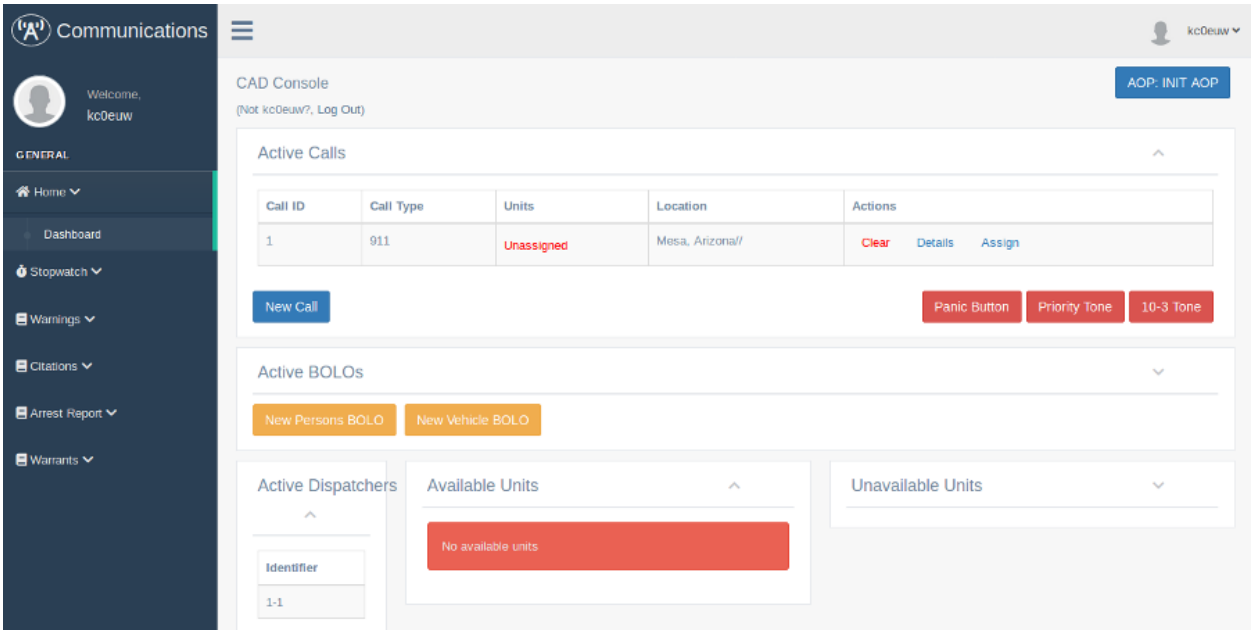



## 20.3 OpenCAD

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 |

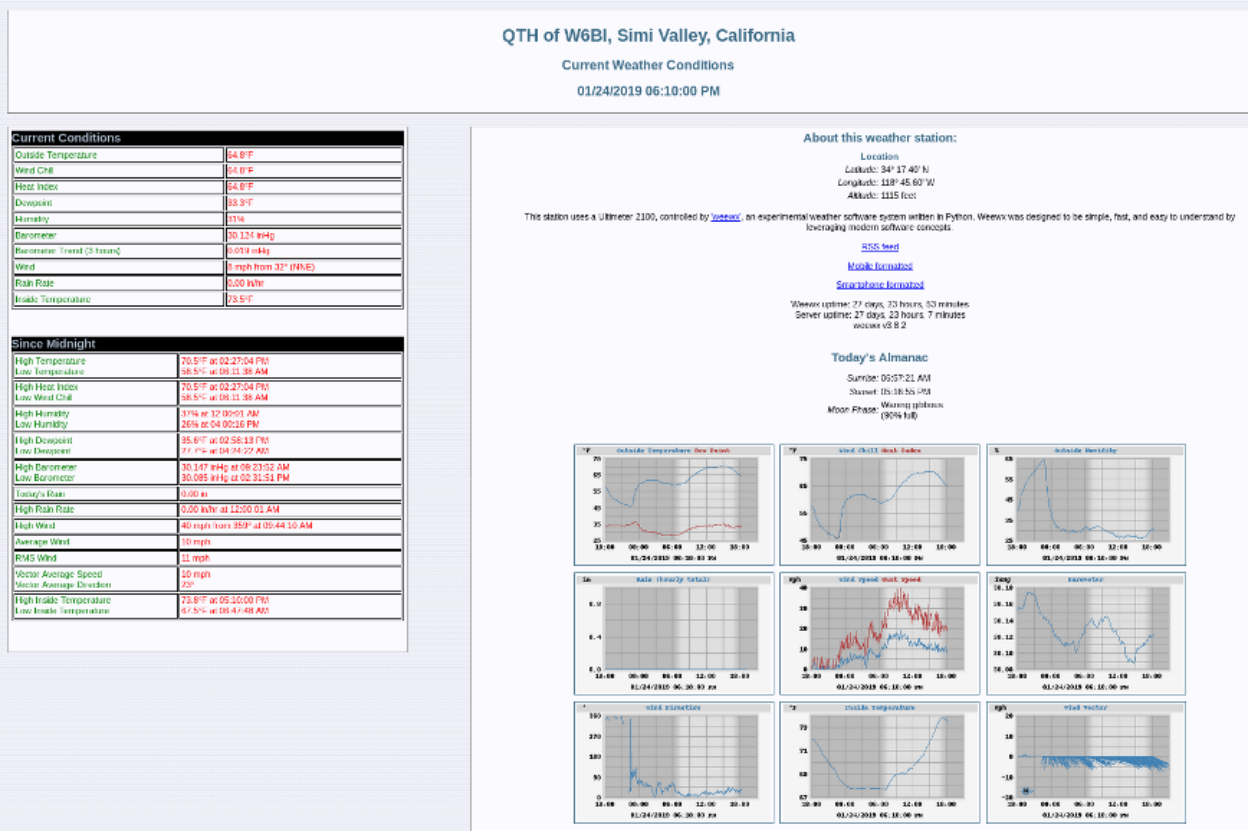
# CHAPTER 21

# 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.



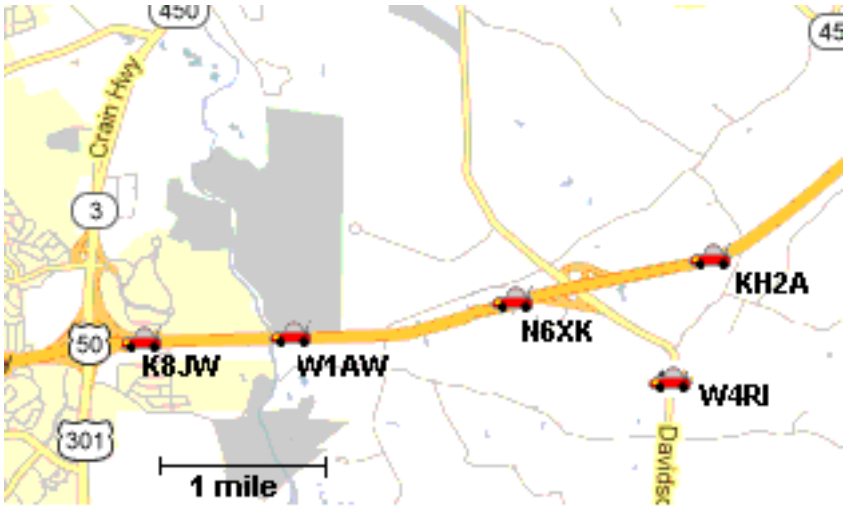

## 21.2 Network Time Services

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

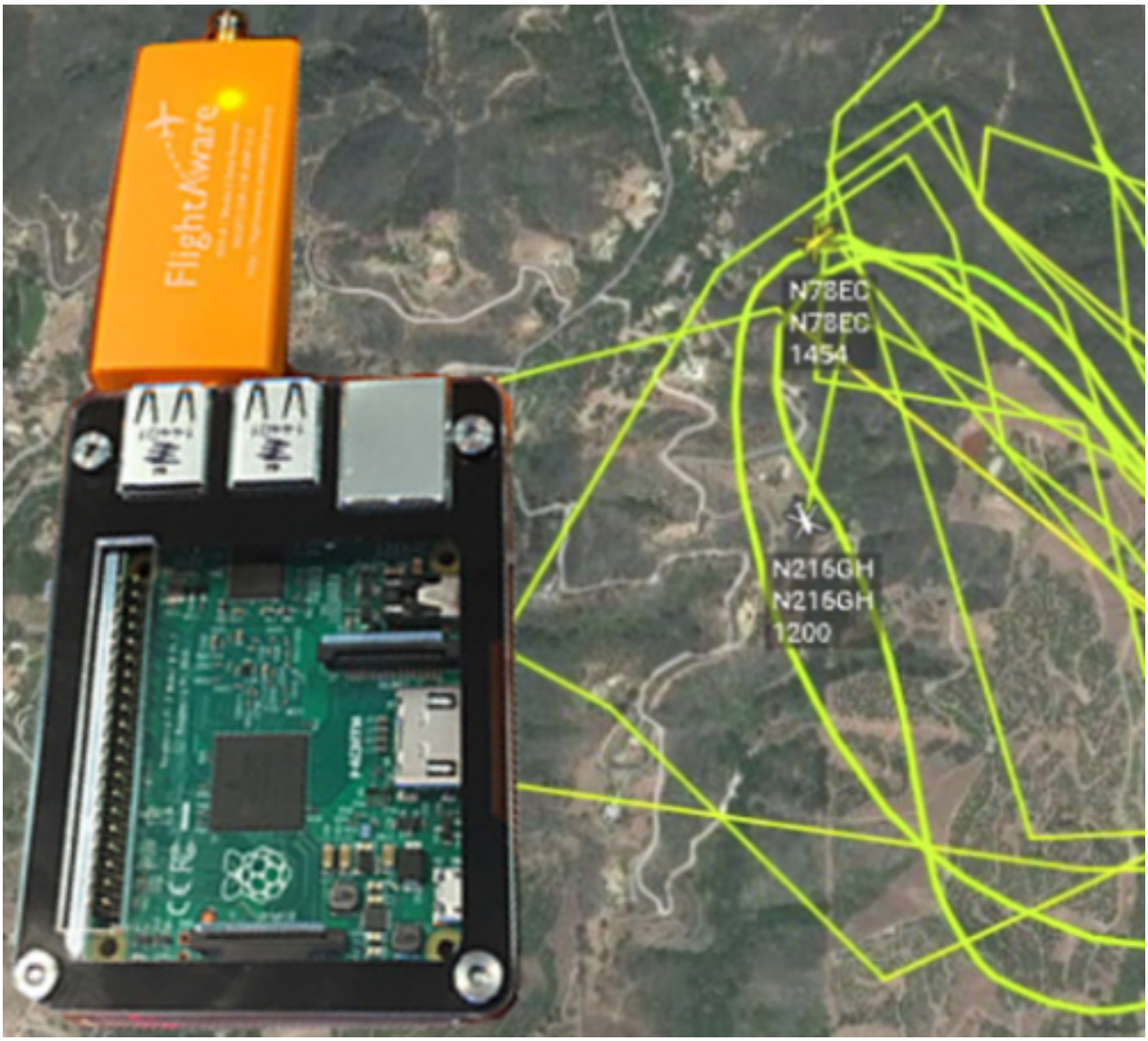



Tracking deployed resources is an important task during any emergency. There are many options for monitoring and displaying the GPS locations of tracked resources, two of which are mentioned here.

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.

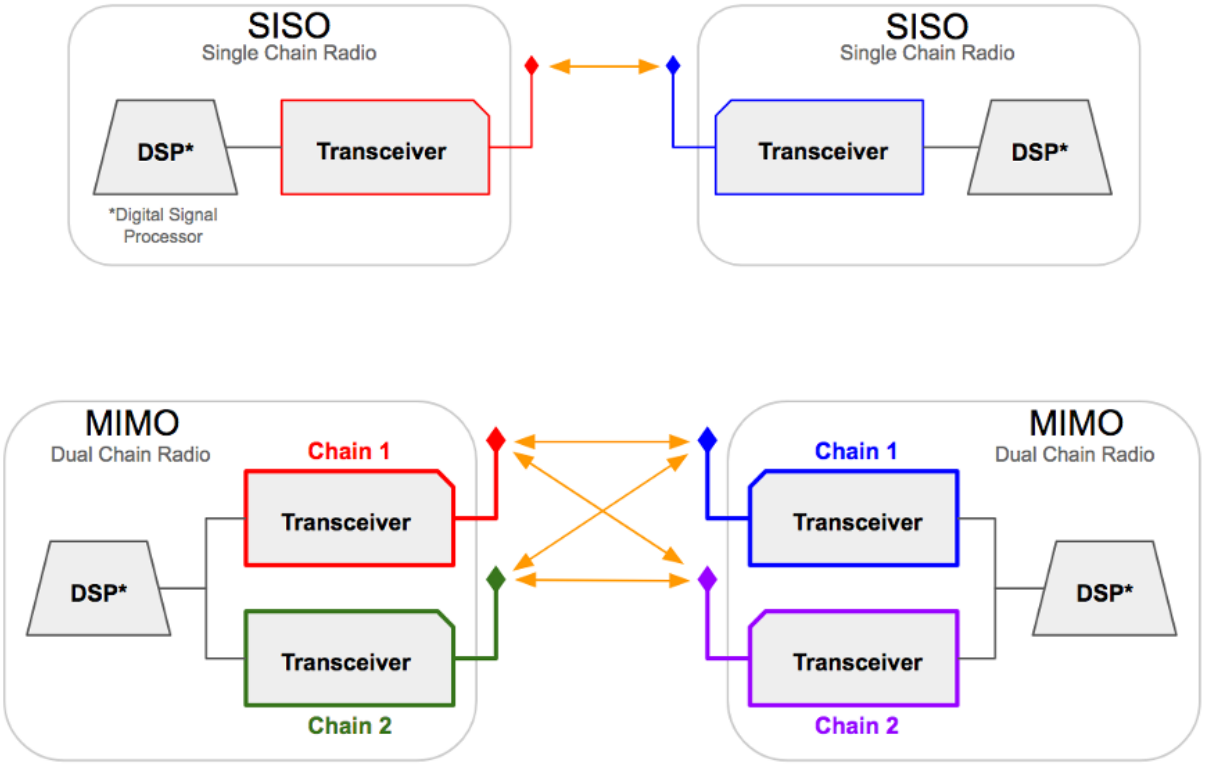
When using a web browser to perform an upgrade, be sure to clear the browser's cache to remove any cached pages remaining from your node's previous firmware version. A clear cache will help to eliminate confusion when displaying node data in the browser.

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

# Comparing SISO and MIMO Radios

SISO (Single Input Single Output) radios have a single transceiver-antenna chain, while MIMO (Multiple Input Multiple Output) devices have multiple chains coordinated through the Digital Signal Processor (DSP). The MIMO devices supported by AREDN® have dual chains for both transmit and receive, and they support dual data streams [2x2:2].



Both SISO and MIMO radios use OFDM (Orthogonal Frequency Division Multiplexing), which inherently handles poor RF conditions such as multipath interference or fading. The rate selection algorithm in the wireless driver adapts to changing RF conditions so that the optimal MCS rate is always used. The selected MCS includes the appropriate modulation, forward error correction, and number of data streams.

## 23.1 SISO Radios

By design SISO radios transmit all of their RF power on a single polarization. While it may seem like an advantage to have full power concentrated on a single polarization, there are specific limitations to SISO devices. A single chain device can only transmit one data stream at a time, and SISO devices do not have the ability to process and enhance multiple signals received simultaneously.

SISO radios are also limited in the data throughput they can achieve on their single chain. For example, a SISO device is limited to the 802.11n MCS7 (Modulation and Coding Scheme) protocol rate of 32.5 Mbps with Long Guard Interval (LGI), while a MIMO device using MCS15 (Modulation and Coding Scheme) can achieve up to 65 Mbps. In this regard SISO is at a definite disadvantage since it lacks sophisticated signal combining and the multiple simultaneous data streams that are possible with MIMO.

## 23.2 MIMO Radios

One of the advantages of MIMO radios is their ability to exploit multipath signals, achieving a better Signal to Noise Ratio (SNR) by combining multiple received transmissions. This is accomplished using 802.11n technologies such as Polarization Diversity and Maximal Ratio Combining.

On MIMO devices the total transmit power is split between its two polarizations, which means that MIMO radio signals have lower EIRP per polarization. It is possible that SISO radios on both ends of a link could have SNR values that match those of MIMO radios using 802.11n MCS0 (Modulation and Coding Scheme) to MCS7 on that same link. However, a MIMO device using MCS0 to MCS7 will transmit its data stream on both chains simultaneously, providing a distinct advantage on the receiving end where the MIMO radio uses MRC to enhance the signal. MRC is used when multiple antennas receive the same data stream, which applies only for MCS0 to MCS7. With MCS8 to MCS15 Spatial Multiplexing achieves multiple simultaneous data streams.

Given the same channel width and link characteristics, MIMO tends to out-perform SISO in both reliability and throughput. A good test to verify this would be to compare the performance of SISO vs. MIMO radios between the same endpoints. MIMO radios can attain double the throughput because they are capable of using twice the MCS rate. In the final analysis, the technology limitations of SISO will not allow it to match the throughput levels that are possible with MIMO.

## 23.3 SISO - MIMO Combinations

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

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


---

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


---

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


---

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


## 23.4 Troubleshooting Tips

- Whenever possible try not to mix device types on radio links. As a general rule, use MIMO-to-MIMO for most types of radio links.
- If you have a marginal SISO-to-SISO link and you must replace one of the radios, either install another SISO radio or replace both ends with MIMO devices. A marginal but usable link between SISO devices may become unusable if one is replaced with a MIMO device.

Additional information on the operation of SISO and MIMO radios can be found in references such as this: MIMO for Dummies.

CHAPTER 24

# 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.

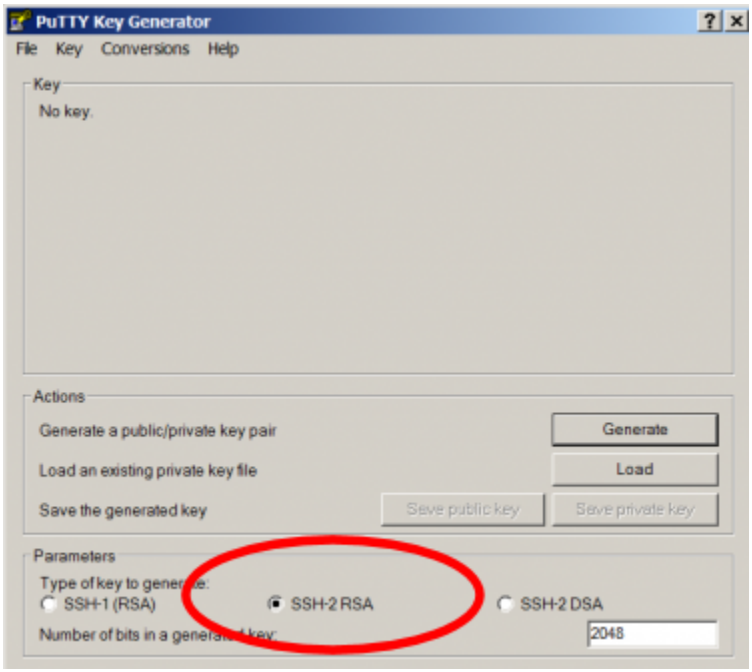
The use of Secure Shell (SSH) keys when using PuTTY or another SSH client is a useful aid to managing a group of AREDN® nodes.

First, obtain the PuTTY suite of applications from the PuTTY Download Page and install them on your computer.

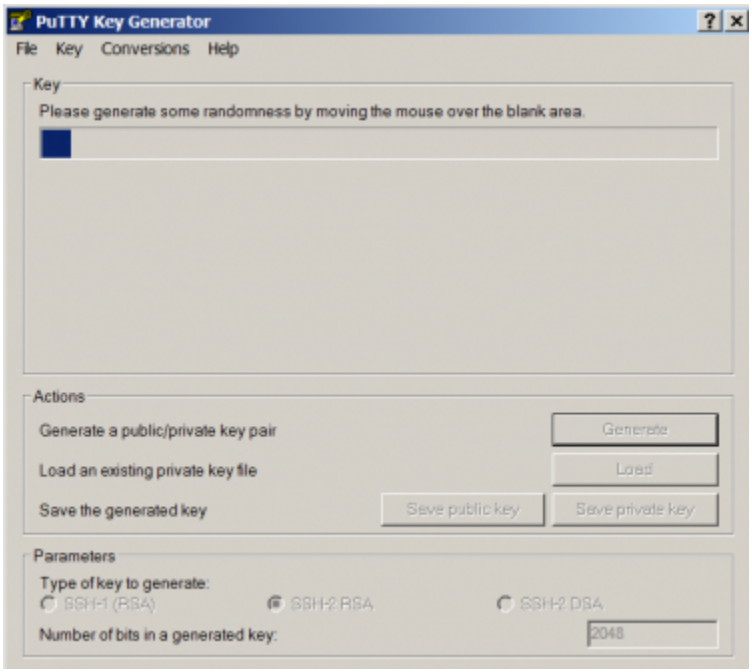
Second, obtain and prepare to use a text editor such as Notepad++ that does not insert unwanted characters and metadata into a text file.

Next, follow the steps below.

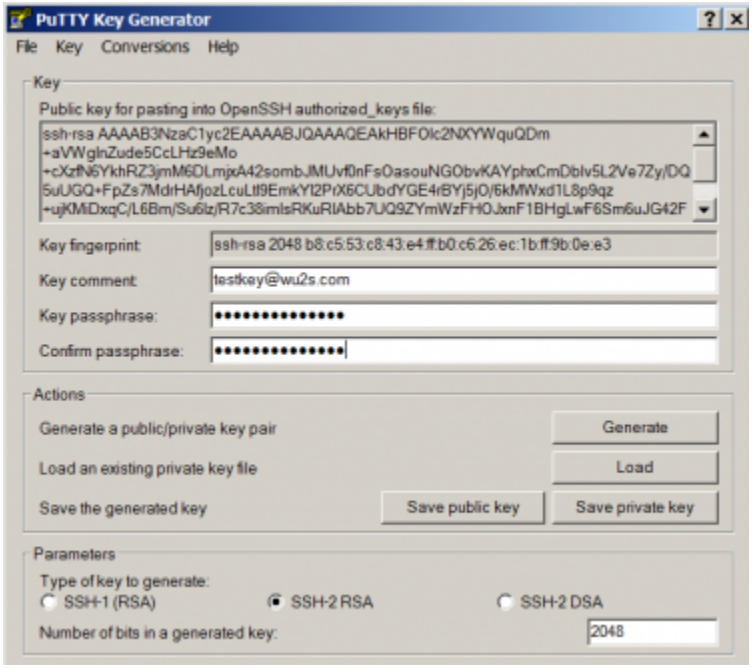
1. Start the PuTTYGen application. Confirm that you are going to generate an SSH-2 RSA key.



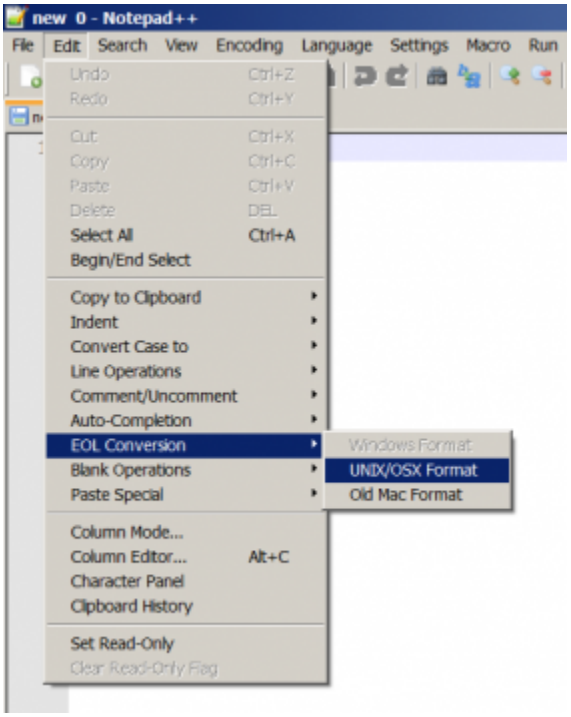

2. Select the *Generate* button to get the prompt asking you to make some random mouse movements. After a short while you get a message asking you to wait while the keys are generated. It finishes and you now have a new key pair.



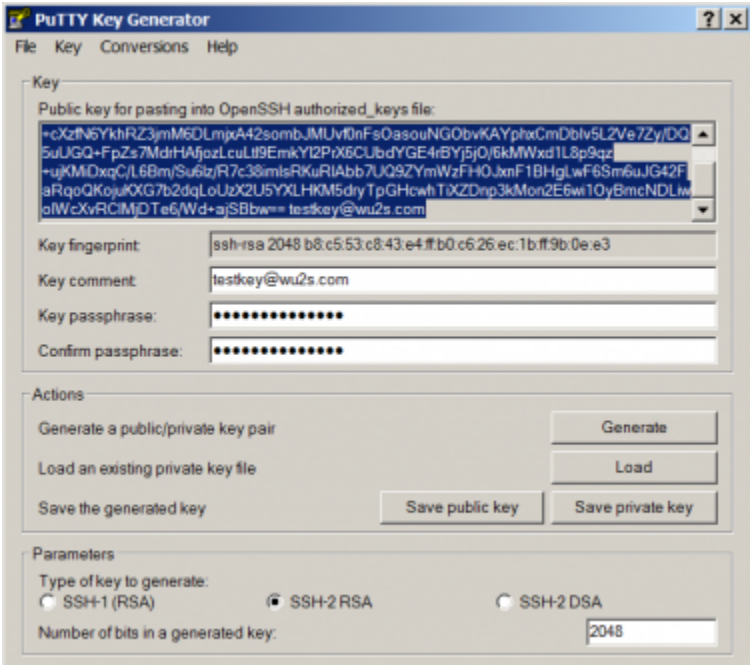

3. Give the key pair a suitable comment so that you will remember what the keys are used for. Here we just entered testkey@wu2s.com for an example. Whatever you enter in the "Key Comment" field must look like an email address with no spaces and the "@" present as in *callsign@example.com*. Also enter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



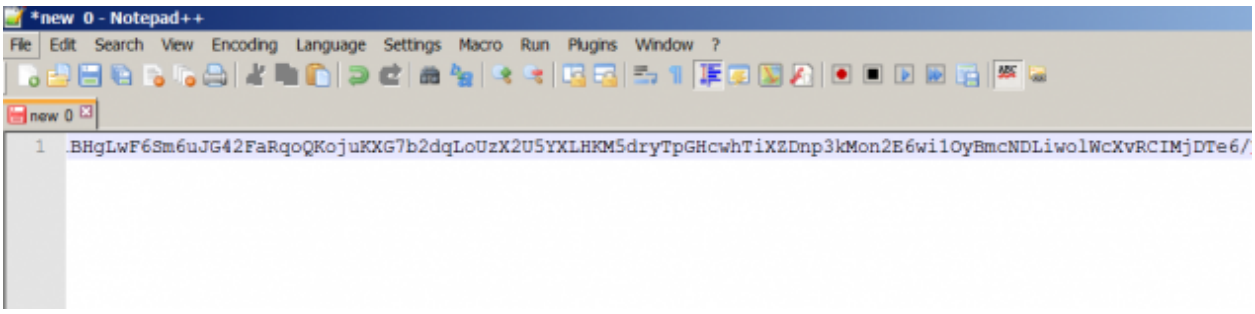

4. Now copy and save the public key. Open Notepad++ and confirm that the End Of Line (EOL) format is set to UNIX/OSX Format. This will ensure that there are no extraneous characters in the public key file.



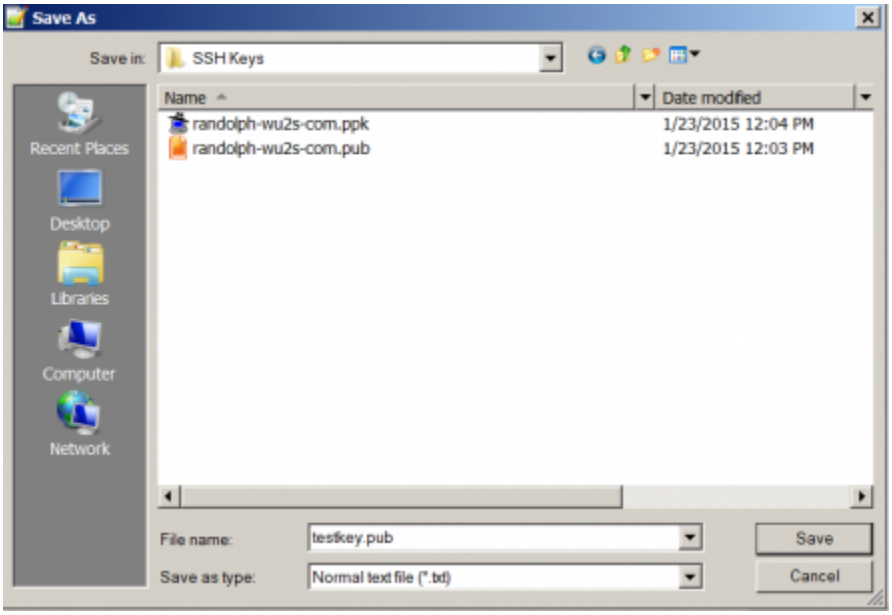

5. Back in your PuTTYGen window, select and copy (Control-C) the complete text in the boxed labeled “Public key for pasting into OpenSSH authorized keys file”



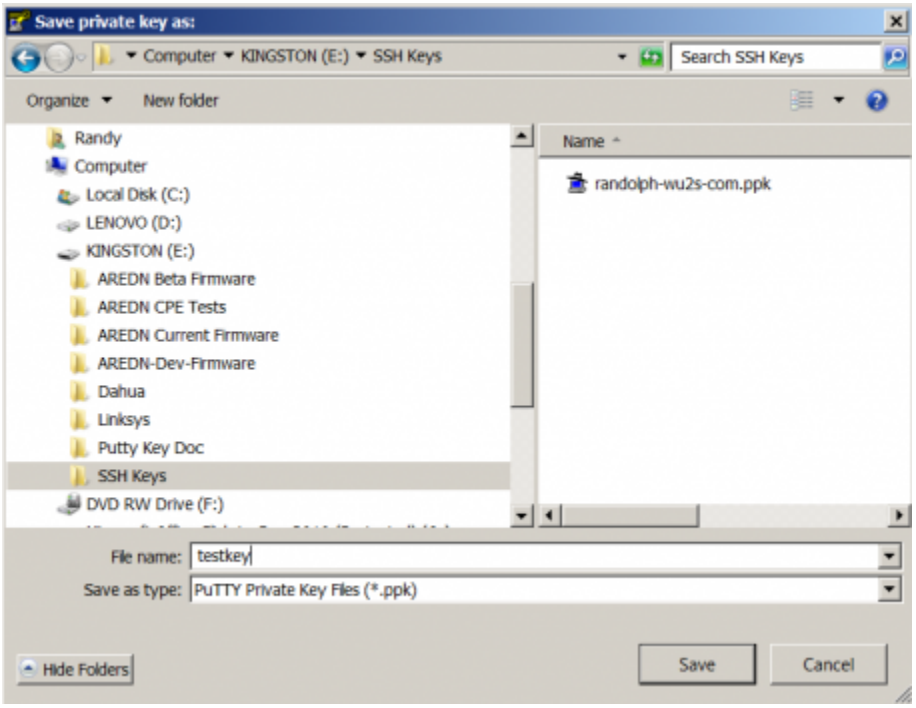

6. Switch back to your Notepad++ window and Paste (Control-V) the public key text you just copied from PuTTYGen.



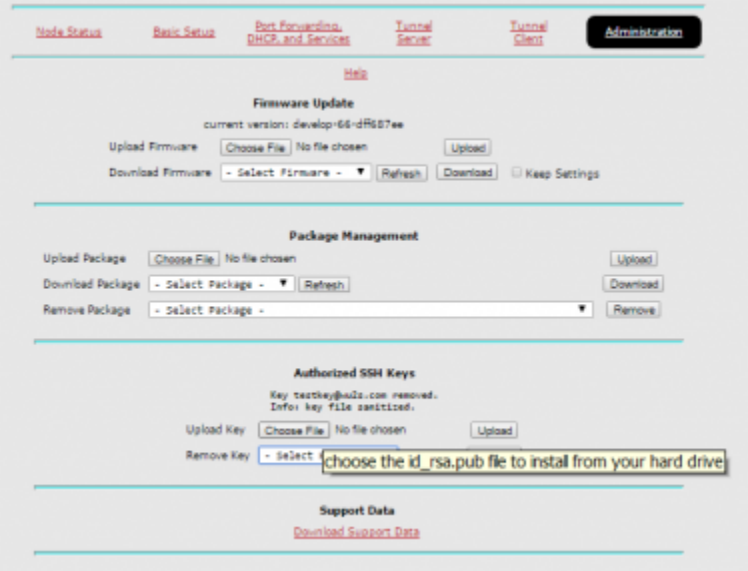

7. From the Notepad++ menu bar, select File -> Save As to save the public key to a suitable location. Many people save their keys on a USB flash drive to maintain physical possession of them at all times. Give the public key file a suitable name. You can exit Notepad++ now since you will not need it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



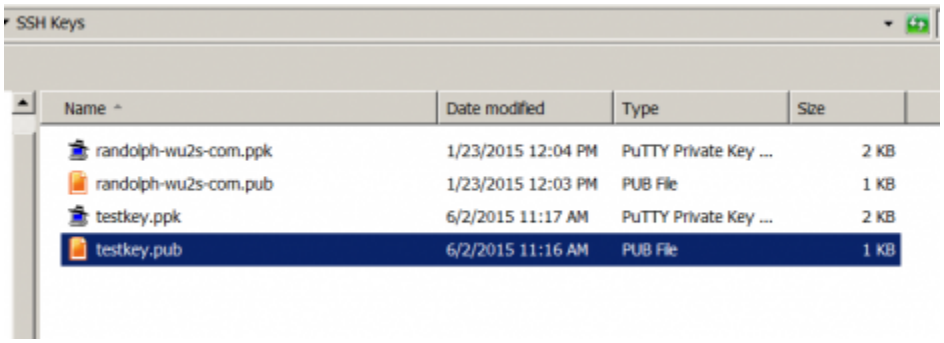

8. Switch back to the PuTTYGen window again and select the "Save Private Key" button. This will let you save the private key just as you did in the previous step with the public key. You are finished generating and saving your SSH keys. Exit PuTTYGen.



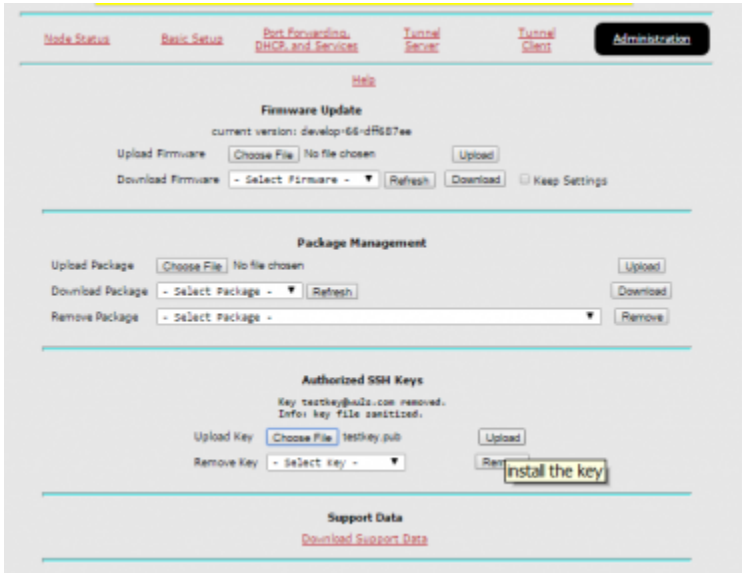

9. In order to use your new SSH key pair, login to your AREDN® node and go to the *Setup -> Administration* screen. At the bottom you will see the Authorized SSH Keys section where you will install the public keys to use on this node.



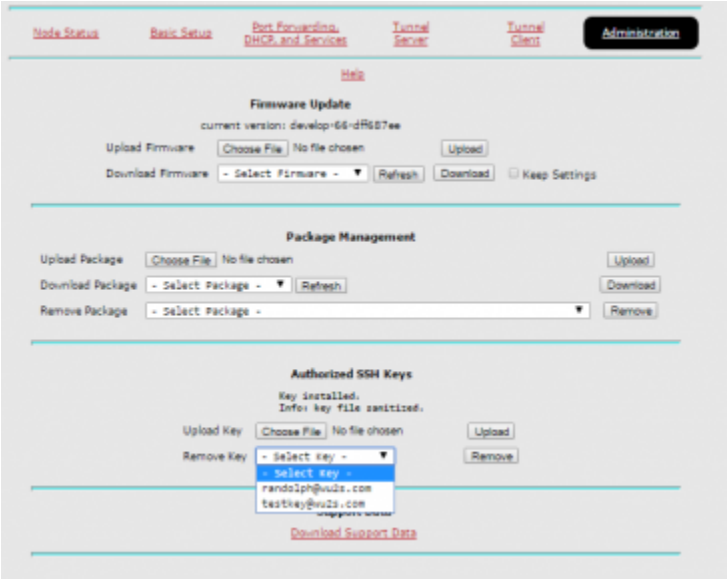

10. When you press the Select File button you see a dialog box which enables you to locate the public SSH key that you want to install.



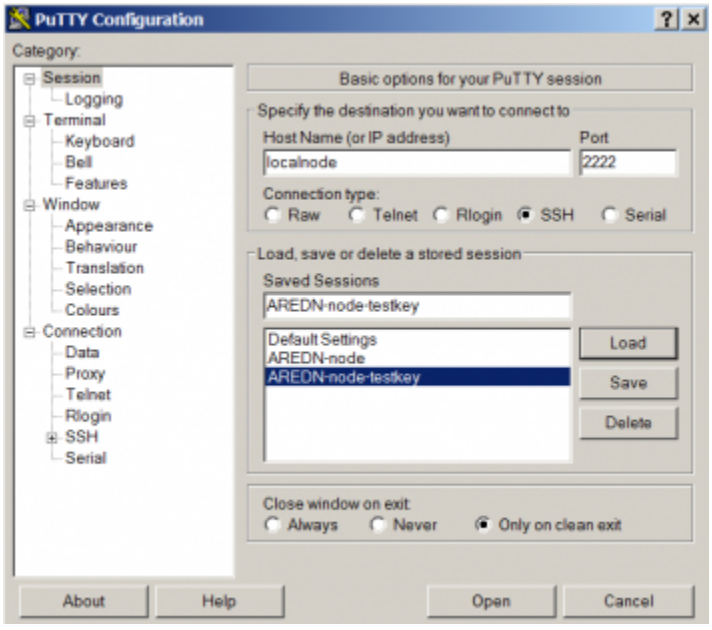

11. After choosing the desired public key file. Select the *Upload* button to install the key on the AREDN® node.



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.

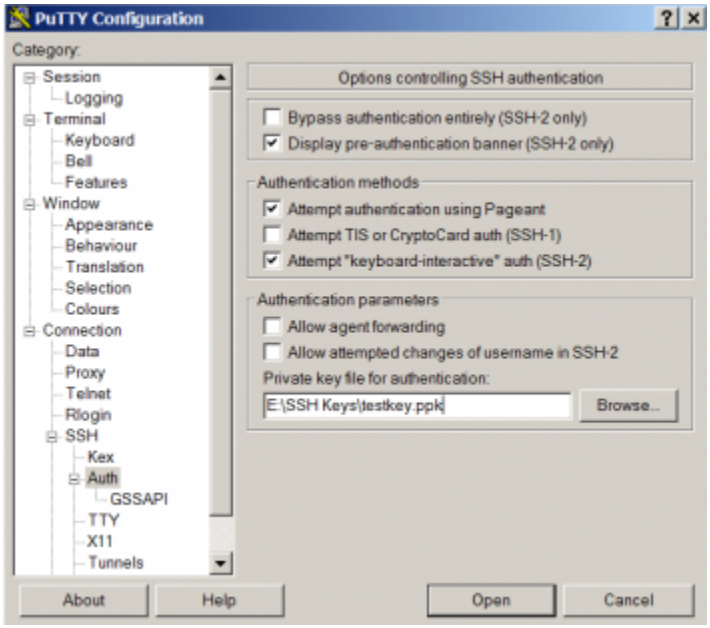



13. To use your SSH keys, open a new PuTTY session. In the Hostname box enter *localnode* and in the Port box enter 2222. It is helpful to save this session definition as something you will remember. Select the Save button.

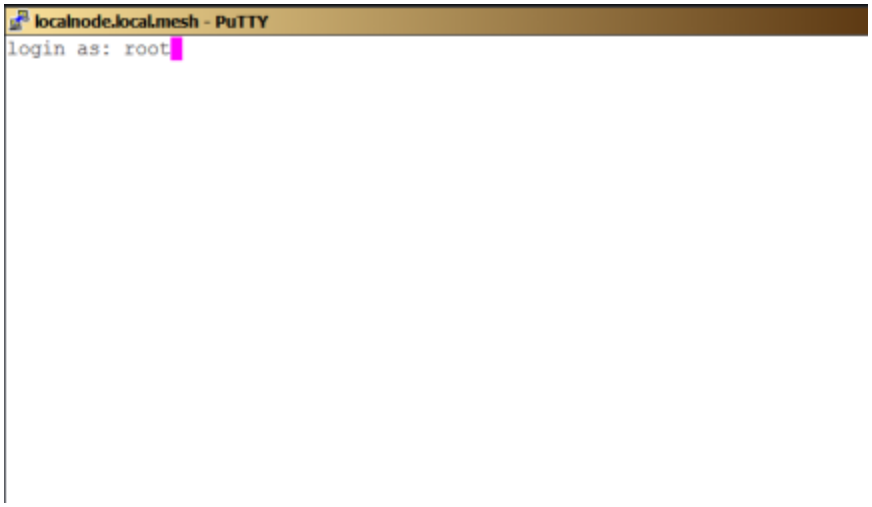



14. Now, using the menu at the left, go to the SSH section and then select the *Auth* item. This shows a number of Options. The only one we need is the very last – the location of the Private key file for authentication. Browse for it and select the correct filename as before. Remember that the PRIVATE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

the session definition again.

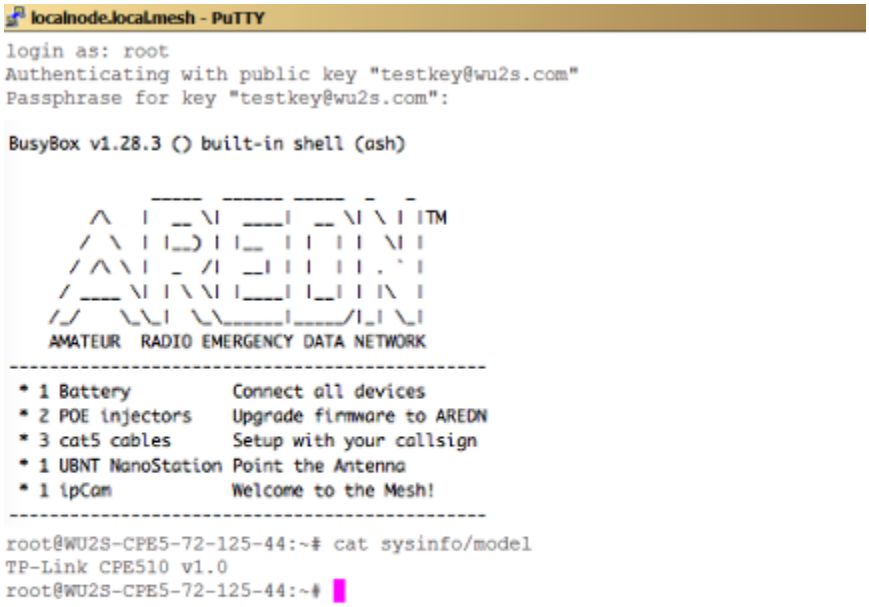



15. Now you can use the session information you saved by clicking the *Load* or *Open* button in the main PuTTY session screen. This will open a terminal session box as shown below. Login to the AREDN® node as *root*.




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.


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.



# CHAPTER 25

# 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 ($\mu$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 $\mu$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 26

# 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.

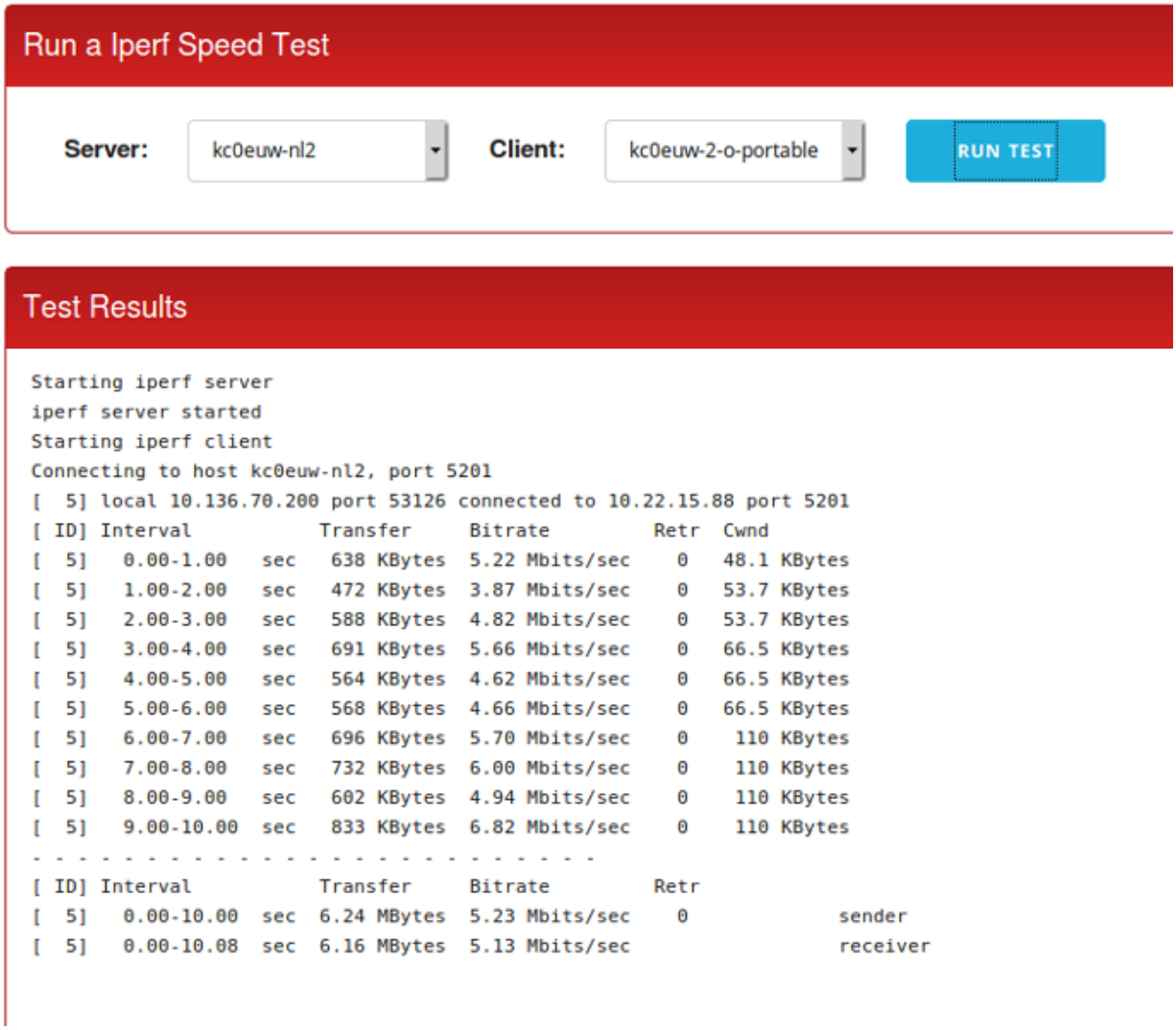
## 26.1 Installing iperf and IperfSpeed

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.

## 26.2 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.




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 27

# Tools for Developers

This section of the AREDN® documentation contains information useful for developers who want to retrieve information from one or more nodes for use in any of several applications. For example, a developer may want to write a program which periodically polls a set of nodes to gather link quality or signal values to insert them into a network management or historian system for trending and analysis. The popular KG6WXC MeshMap application uses these tools to create and update a comprehensive mesh network map.

## 27.1 SYSINFO.JSON

The **sysinfo.json** API (Application Programming Interface) has been included in AREDN® firmware for several releases, and each release includes an *api_version* tag which can be used to track the feature set supported by that version of the API. As new features are added, the *api_version* number is incremented.

The basic API retrieves general node information in JSON format, and it can be invoked using the following URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json`

The following information is always returned in the JSON data stream:

- Node name
- API version
- Latitude, longitude, and grid square (if available)

- *Node Details* section containing the firmware manufacturer and version, the radio model and board ID, and the node description text (if any)
- *Sysinfo* section containing node uptime and load averages for the last one, five, and fifteen minutes
- *Interfaces* section containing the name, MAC address, and IP address (if any) assigned to each of the node's network interfaces
- *Mesh RF* section containing the SSID, channel, center frequency, channel width, and status of the mesh radio
- *Tunnels* section showing whether the tunnel package is installed and the number of active tunnels (if any)

The values returned by the API are represented in the following snippet of raw JSON. This is only a sample of the full data stream containing all of the values described above.


```
{
"api_version": "1.7",
"lat": "33.101010",
"lon": "-101.101010",
"grid_square": "DM22xx",
"node": "CALLSIGN-NODE-22",
"sysinfo": {
  "uptime": "5 days, 6:22:30",
  "loads": [
    0.05003,
    0.05003,
    0
    ]
  },
"node_details": {
  "description": "CALLSIGN-22 node information here...",
  "model": "MikroTik RouterBOARD 952Ui-5ac2nD ",
  "board_id": "0x0000",
  "firmware_mfg": "AREDN",
  "firmware_version": "1101-ad0caaf"
  }
}
```


In addition to the basic information described above, which is always returned with every invocation, the **sysinfo.json** API can also include other details based on the flags appended to the URL as explained below. In some cases it may be useful to include more than one of the following flags in the URL, and these flags can be combined using the `&` operator. For example, `sysinfo.json?hosts=1&services=1` will include both the *hosts* and *services* information in addition to the basic details which are always returned.

### 27.1.1 Add Hosts Information

To retrieve mesh hosts information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?hosts=1`

A *hosts* section will be included in the JSON data stream containing an entry for each node and mesh-connected device. The *name* and *IP* address of each device will be shown. The values returned by the *hosts* flag are represented in the following snippet of raw JSON.

```
...
"hosts": [
  {
    "name": "CALLSIGN-NODE-22",
    "ip": "10.22.22.22"
  },
  {
    "name": "CALLSIGN-VOIP-PHONE",
    "ip": "10.22.22.24"
  },
  {
    "name": "MYCALL-NODE-81",
    "ip": "10.81.81.81"
  },
  {
    "name": "MYCALL-RPI",
    "ip": "10.81.81.83"
  }
],
...
```

### 27.1.2 Add Services Information

To retrieve mesh services information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?services=1`

A *services* section will be included in the JSON data stream containing an entry for each service available on the mesh. Each entry will include the service *name*, *protocol*, and *link* URL. The values returned by the *services* flag are represented in the following snippet of raw JSON.

```
...
"services": [
  {
    "name": "IperfSpeed",
    "protocol": "tcp",
    "link": "http:\/\/MYCALL-NODE-81\/iperfspeed"
```

(continues on next page)

(continued from previous page)

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

### 27.1.3 Add Local Services Information

To retrieve information about the services provided only through a single node, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?services_local=1`

A *services_local* section will be included in the JSON data stream containing an entry for each service available through the node being queried. Each entry will include the service *name*, *protocol*, and *link* URL as described above.

### 27.1.4 Add Link Information

To retrieve mesh link information, invoke the API using the following flag on the URL: `http://<nodename>.local.mesh/cgi-bin/sysinfo.json?link_info=1`

A *link_info* section will be included in the JSON data stream containing an entry for each node that is reachable via RF, DTD (Device To Device), or TUN (Tunnel) from the node being queried. Each entry will be identified by the IP address of the reachable node, and within each IP address section you will see the *hostname* (node name), *linkType* (RF, DTD, or TUN), *linkQuality*, *neighborLinkQuality*, *signal*, *noise*, *olsrInterface* name, *tx_rate*, and *rx_rate*. The values returned by the *link_info* flag are represented in the following snippet of raw JSON.

```
...
"link_info": {
  "10.22.22.22": {
    "hostname": "CALLSIGN-NODE-22",
    "linkType": "RF",
    "linkQuality": 0.9543000000,
    "neighborLinkQuality": 0.9748576110,
```

(continues on next page)

(continued from previous page)

```
    "signal": -76,
    "noise": -95,
    "olsrInterface": "wlan0",
    "tx_rate": 6,
    "rx_rate": 4
  },
  "10.81.106.77": {
    "hostname": "MYCALL-NODE-81",
    "linkType": "DTD",
    "linkQuality": 1,
    "neighborLinkQuality": 1,
    "olsrInterface": "eth0.2"
  }
},
...
```

# CHAPTER 28

# Frecuencias y canales

Las frecuencias y canales disponibles para la red AREDN® se muestran a continuación.

### 2.4 GHz

| 2.4 GHz | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Canal | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Estado | Banda *HAM* | | | Compartido con bandas *HAM* y *Wi-Fi/ISM* | | | | | |
| Frecuencia | 2.397 | 2.402 | 2.407 | 2.412 | 2.417 | 2.422 | 2.427 | 2.432 | 2.437 |

### 3.4 GHz

| 3.4 GHz | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Canal | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 |
| Estado | Banda *HAM* | | | | | | | | | | | |
| Frecuencia | 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 |
| Banda *HAM* | | | | | | | | | | | |
| 3.440 | 3.445 | 3.450 | 3.455 | 3.460 | 3.465 | 3.470 | 3.475 | 3.480 | 3.485 | 3.490 | 3.495 |

Para coordinación, consulta tu *Plan de bandas*

### 5.8 GHz

| 5.8 GHz | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Canal | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 |
| | Estado | Banda *HAM* compartida con *U-NII-2C/Wi-Fi/No licenciada* | | | | | | | | | | | | |
| | Frecuencia | 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 | |
| | Banda *HAM* compartida con *U-NII-3/Wi-Fi/No licenciada* | | | | | | | | | | | | | |
| | 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 | |
| | Banda *HAM* compartida con *U-NII-3/Wi-Fi/No licenciada* | | | | | | ** Ver nota al pie* | | | | | | Banda *HAM* | |
| | 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 | |
| | Banda *HAM* | | | | | | | | | | | | | |
| | 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 | |

Para coordinación, consulta tu *Plan de bandas*

* Desde 5.825 hasta 5.850 la banda se comparte con proveedores de acceso inalámbrico a Internet (*Part 15.247*)

# CHAPTER 29

# Información adicional

Puedes encontrar información adicional acerca del proyecto AREDN® en los siguientes enlaces.

- Página principal de AREDN
- Foro oficial de AREDN

## 29.1 Contribuyendo con la documentación de AREDN®

Si estás interesado en contribuir con cualquier aspecto del proyecto en auge AREDN®, puedes hacerlo fácilmente a través de GitHub. Para contribuir con el proyecto AREDN®, primero debes registrarte en GitHub, algo gratuito y sencillo.

1. Accede a GitHub desde tu navegador
2. Desde esa misma página podrás registrarte, proporcionando un nombre de usuario (como sugerencia, incluye tu indicativo), e-mail y contraseña
3. Una vez estés registrado, podrás acceder a tu cuenta personal a través del botón `Sign In`
4. Navega hasta el repositorio de documentación del proyecto AREDN®, situado aquí
5. Pincha en el botón `Fork` situado en la esquina superior derecha de la página. Acto seguido tendrás una copia de la documentación de AREDN® en tu cuenta personal de GitHub.
6. Realiza un cloneado de la documentación de AREDN® a tu ordenador personal

```
git clone https://github.com/{TuNombreDeUsuario}/documentation_es
```

7. En tu ordenador personal, navega hasta el directorio donde se encuentra la documentación

```
cd documentation_es
```

Este directorio contiene una copia de la documentación de AREDN®, por lo que toda la edición y añadidos realizados a la documentación se tendrán que hacer dentro de este directorio y subdirectorios relacionados

El ciclo de trabajo para contribuir con la documentación es idéntico al de la contribución de código, descrito en Cómo usar GitHub con AREDN. Podrás encontrar más información más detallada sobre cómo contribuir con el proyecto AREDN®.

La única diferencia reside en el nombre del repositorio `aredn/documentation_es` y de la rama principal `master`. A la hora de nombrar tu rama local, puedes elegir el nombre que más cómodo te resulte. La documentación de AREDN® está escrita y estructurada utilizando lenguaje markdown. El texto se almacena en ficheros `.rst`. Antes de realizar una `pull request`, asegurate de que tus ficheros `.rst` mantienen el formato y renderizan correctamente.

Después de realizar una `pull request`, el equipo de AREDN® lo revisará del mismo modo que se haría con el nuevo código. Una vez que tu aportación es aplicada al repositorio maestro, una operativa en segundo plano (`Webhook`) procesará los cambios, haciéndolos disponibles para lectura y exportación en `Read the Docs`