Appendix A – LibreMesh Installation

The following is the documentation from LibreMesh's Quick Starting Guide (http://libremesh.org/docs/en_quick_starting_guide.html):

Compatible Hardware

It is recommended that the router has at least 8 MB of flash memory. For 4 MB routers use the special firmware named `-mini`.

These wireless routers have been tested with LibreMesh and have 8 MB of flash memory:

- TP-Link WR842ND
- TP-Link WR1043ND
- TP-Link WDR3500
- TP-Link WDR3600
- TP-Link WDR4300
- Dragino MS14
- Alix 2d2
- Ubiquiti Unifi AP
- Ubiquiti AirRouter
- Ubiquiti AirGateway
- Ubiquiti NanoStation M5 XW
- Ubiquiti NanoStation M5 XM
- Ubiquiti NanoBridge M5
- Ubiquiti NanoStation LoCo M2
- Ubiquiti PicoStation M2
- Ubiquiti Bullet M2

Also models with 4 MB have been tested, using the `-mini` images:

- TP-Link WR740N
- TP-Link WR741ND
- TP-Link WR841ND

For detailed information on these routers check out our http://libremesh.org/docs/hardware/index.html page.

Many other models are supported even if not tested yet (as far as we know), see through our firmware images at http://downloads.libremesh.org/dayboot_rely/17.06/targets/.

Get the Firmware

Choose a source for your firmware

- For a precompiled firmware with default configuration (e.g. wireless AP name LibreMesh.org) you can use our http://downloads.libremesh.org/dayboot_rely/17.06/targets/ site;
- You can compile LibreMesh firmware on your computer using https://github.com/libremesh/lime-sdk (advanced, supports custom community profiles);
- Our https://chef.libremesh.org/ platform allows you to build an image of the firmware online. The platform supports custom community profiles.

For more options check the http://libremesh.org/getit.html page.

Download the correct firmware image

Find the download for your router by name or model number. You may need to search different variations or aliases. You can find more detailed router model instructions at https://wiki.lede-project.org/toh/start.

If you are installing for the first time (a router with stock firmware), choose the link ending with `-factory.bin`. If there's no `-factory.bin` image or you are upgrading an existing install of LibreMesh, OpenWrt or LEDE, choose the link ending with `-sysupgrade.bin`.

Installation Procedure

Open your router web interface

Using an ethernet cable connect to a LAN port on your router. Make sure that the ethernet cable is the only active network interface on your computer (e.g. disable the wireless interface).

If the router is running stock firmware, follow manufacturer instructions to connect to the router. Its IP should be written on the original box or under the router. Usually just opening 192.168.0.1 or 192.168.1.1 in a web browser lets you reach the router web interface. If you can't connect to the router because you can't find its IP, you can try the IP address of your gateway. For getting it go to the terminal and use netstat -rn (mac), or ip route show default (Linux). More details on finding your router's IP can be found at http://www.howtogeek.com/233952/how-to-find-your-routers-ip-address-on-any-computer-smartphone-or-tablet/ and http://www.computerworld.com/article/2474776/wireless-networking/network-security-find-the-ip-address-of-your-home-router.html.

If the router is running OpenWrt or LEDE the instructions in the previous paragraph should apply.

If the router is running LibreMesh just opening http://thisnode.info should get you to the web interface.

Then you can log in as admin (if unchanged, the username and password will be on the router's box, on OpenWrt by default there's no admin password, on LEDE by default it's an empty password).

For more connection options see http://libremesh.org/docs/en_connecting_nodes.html.

If you suspect you can't connect to your router because of a damaged configuration, follow the Troubleshooting guide at http://libremesh.org/docs/en_troubleshooting.html.

Flashing

For Ubiquiti AirMax series routers, flashing on top of AirOS versions 5.6.x will brick your device (the recovery procedure requires opening the router chassis and connecting directly to its serial port). If your router has AirOS 5.6.x, you will have to find and download an AirOS 5.5.x version and use it to downgrade your router.

Once you've logged in as root or admin in your router, reach the firmware upgrade page.

If there's a Keep Settings option, take care to UNCHECK it. It is checked by default on OpenWrt/LEDE.

Upload the firmware image file you've downloaded and click Flash Image. Wait a couple of minutes for the process to complete. Reconnect to the ethernet interface (for getting the new IP) and open http://thisnode.info.

Congratulations, you have a working LibreMesh router!

Connect to Your LibreMesh Router

Just connect to the router via its wireless AP interface or via ethernet cable on its LAN port and open http://thisnode.info in the web browser.

If is the first time you connect to the router, you will have to set an admin/root password. Leaving a LibreMesh router with no admin password is a huge security risk.

For more connection options see How to connect to nodes page at http://libremesh.org/docs/en_connecting_nodes.html.

If you suspect you can't connect to your router because of a damaged configuration, follow the Troubleshooting guide at http://libremesh.org/docs/en_troubleshooting.html.

Share the Internet Connection with the LibreMesh Network

LibreMesh is automatically sharing with the rest of the mesh network any internet connection is connected to the router WAN port. There's no problem if more than one internet gateway is connected to the LibreMesh network, likely the one closest to the client will be used.

If the LibreMesh router has no WAN port (just LAN ports, or just one ethernet port), one of the ethernet ports has to be configured as WAN port in order to share the internet connection. Refer to next section for configuration.

Configuration

WORK IN PROGRESS

Refer to the LibreMesh config file page for detailed information at http://libremesh.org/docs/en_config.html.

Using the Console Interface (optional)

Until here we went through the installation, connection and configuration procedures using the LibreMesh web interface.

As in every Linux-based system there's the availability of a textual console interface for advanced configuration and hardcore users.

This part of the guide should not be needed for normal LibreMesh use.

Flashing Via the Console Interface (optional)

This is possible just if you're upgrading an existing OpenWrt, LEDE or LibreMesh installation, not from stock firmware.

Copy the downloaded firmware image to the /tmp directory on your target router using the scp command.

Do not try to copy the firmware image to directories different from /tmp. They have limited memory access.

In case the router already has LibreMesh you can do this with

scp /LOCAL/PATH/TO/BUILD.bin root@thisnode.info:/tmp/

Otherwise (upgrading from OpenWrt or LEDE) you will need to insert the router IP address in

scp /LOCAL/PATH/TO/BUILD.bin root@ROUTERIPADDRESS:/tmp/

When upgrading from OpenWrt, in order to connect via ssh/scp you will need to have an admin/root password set via the web interface or via telnet and the passwd command. Check out Connecting to your own node for help on this at http://libremesh.org/docs/en_connecting_nodes.html.

Now connect to the console interface using ssh, if LibreMesh is already running with ssh root@thisnode.info or with ssh root@ROUTERIPADDRESS if OpenWrt or LEDE are running.

Then enter the /tmp directory where the firmware is present with cd /tmp, check the presence of the file with ls and install it with

sysupgrade -n lede-ROUTERMODEL-squashfs-sysupgrade.bin

The -n option for sysupgrade command is needed for discarding the previous configuration files. Omitting the -n option is never a good idea when flashing LibreMesh.

For more information on the sysupgrade process, see these OpenWRT instructions at http://wiki.openwrt.org/doc/howto/generic.sysupgrade.

Configuring Via the Console Interface (optional)

Rather than using the web interface, modifying directly /etc/config/lime file allows to access more advanced options but increases the risk of writing broken configuration.

You can use the vi or vim text editor for editing /etc/config/lime, the settings in this file will override the default ones in /etc/config/lime-defaults.

You can find examples and documentation in the /docs/lime-example file (you can find it online here at https://github.com/libremesh/lime-packages/blob/develop/packages/lime-docs/files/lime-example) as well as in LibreMesh config file page at http://libremesh.org/docs/en_config.html.

After saving the edits to the lime file, apply the changes to system configuration files launching the lime-config command. Next reboot the router with reboot && exit to apply the new settings.

Appendix B – CJDNS Installation and Configuration on LibreMesh

The older versions of LibreMesh (before 17.06 DaybootRely) do not have the cjdns opkg in the repositories, which is required by the luci-app-cjdns package. In order to install on older versions, the package must be manually transferred and installed to the node.

This installation can be done from the web interface
1. Log in to the terminal via ssh, then update the repositories and install the luci-app-cjdns. This will automatically install the cjdns package as well, which is a prerequisite for the luci app. Optionally, you should install iperf3 for network testing

```
opkg update && opkg install luci-app-cjdns
opkg install iperf3
```

2. Ethernet beaconing is turned on by default for all interfaces.

Appendix C – Home/Client Node – LibreMesh Based

The purpose here is to replace the cable router an ISP would provide in a standard Internet service installation. A home node does not necessarily require LibreMesh if it is not being run on a router.

A LibreMesh-based home node requires:

- Completion of the steps to install LibreMesh on a capable router and configure CJDNS.
- A high-gain outdoor mountable antenna with cable long enough to reach it.
- An UPS dedicated to network equipment.
- (Optional) A unidirectional access point or antenna for the router.

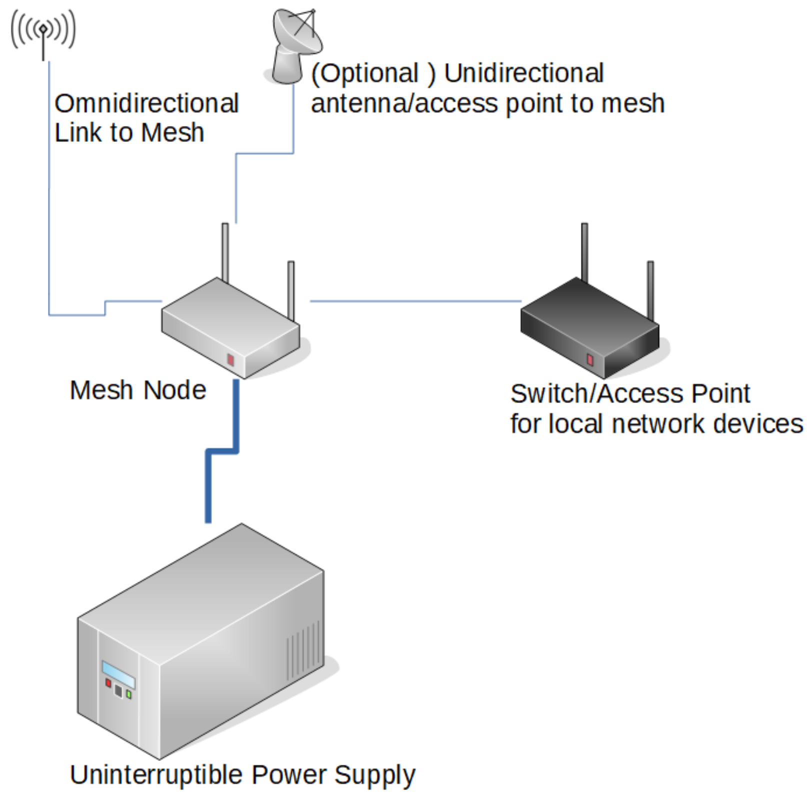

Figure 7: Example Home Node Setup

1. Install the package **kmod-ipt-nat6**

```
opkg update && opkg install kmod-ipt-nat6
```

2. Enter the following ip6tables rules into the file **/etc/firewall.user**:

```
ip6tables -t nat -A POSTROUTING -o tuncjdns -j MASQUERADE
ip6tables -A FORWARD -i tuncjdns -o br-lan -m state --state RELATED,ESTABLISHED -j
ACCEPT
ip6tables -A FORWARD -i br-lan -o tuncjdns -j ACCEPT
```

3. Under Network > Interfaces, click **Edit** next to **LM_NET_BR_LAN_ANYGW_IF.** Scroll down to DHCP Server and click the IPv6 Settings tab. Change DHCPv6-Service to **server mode**, DHCPv6-Mode to **stateful-only**.

4. After rebooting the device, verify that the computer can retrieve an IPv6 address from the access point. To test CJDNS connectivity, ping a CJDNS connected address from the computer.

Appendix D – Home/Client Node – Raspberry Pi Based

This is a home node setup courtesy of the Toronto Mesh Net group. It works on the Raspberry Pi 1, 2, 3, and Orange Pi Zero.

After initial pi setup, the following commands are to be run in the terminal:

```
wget https://raw.githubusercontent.com/tomeshnet/prototype-cjdns-pi/develop/scripts/install /
&& chmod +x install && WITH_MESH_POINT=true WITH_WIFI_AP=true WITH_IPFS=false /
WITH_PROMETHEUS_NODE_EXPORTER=true WITH_PROMETHEUS_SERVER=false WITH_GRAFANA=false / WITH_H_DNS=true
WITH_H_NTP=true WITH_FAKE_HWCLOCK=true WITH_EXTRA_TOOLS=true / TAG_PROTOTYPE_CJDNS_PI=develop ./install

sudo systemctl stop hostapd
sudo systemctl disable hostapd
sudo sed -i s/wlan0/eth0/ radvd.conf
sudo sed -i s/wlan0/eth0/ dnsmasq.conf
sudo sed -i s/eth0/eth1/   /etc/network/interfaces
sudo sed -i s/wlan0/eth0/ /etc/network/interfaces
sudo sed -i s/eth0/tun0/   /etc/hostapd/nat.sh
sudo sed -i 's/^exit 0/\/etc\/hostapd\/nat.sh\nexit 0/' /etc/rc.local
```

The first line will cause the machine to automatically reboot after it has completed running. Then, after entering the rest of the commands, the ethernet interface will provide DHCP to the network and NAT all traffic into CJDNS and through the mesh.

Appendix E – Infrastructure Node

An infrastructure node's primary purpose is to connect nodes that cannot communicate with each other either due to distance or due to topographical obstructions. This set-up is a minimal infrastructure node with one omnidirectional transmitter and one unidirectional transmitter. Power requirements will expand with more transmitters. Infrastructure nodes can be as small using single omnidirectional antenna, using two directional antennas, or as large as 4-directional antennas, a PoE camera, and a Raspberry Pi. All infrastructure node builds should be tailored to a specific need. A small area might only need an infrastructure node to get around a canyon edge, which would only need two directional antennas. A node might only need one strong omnidirectional antenna.

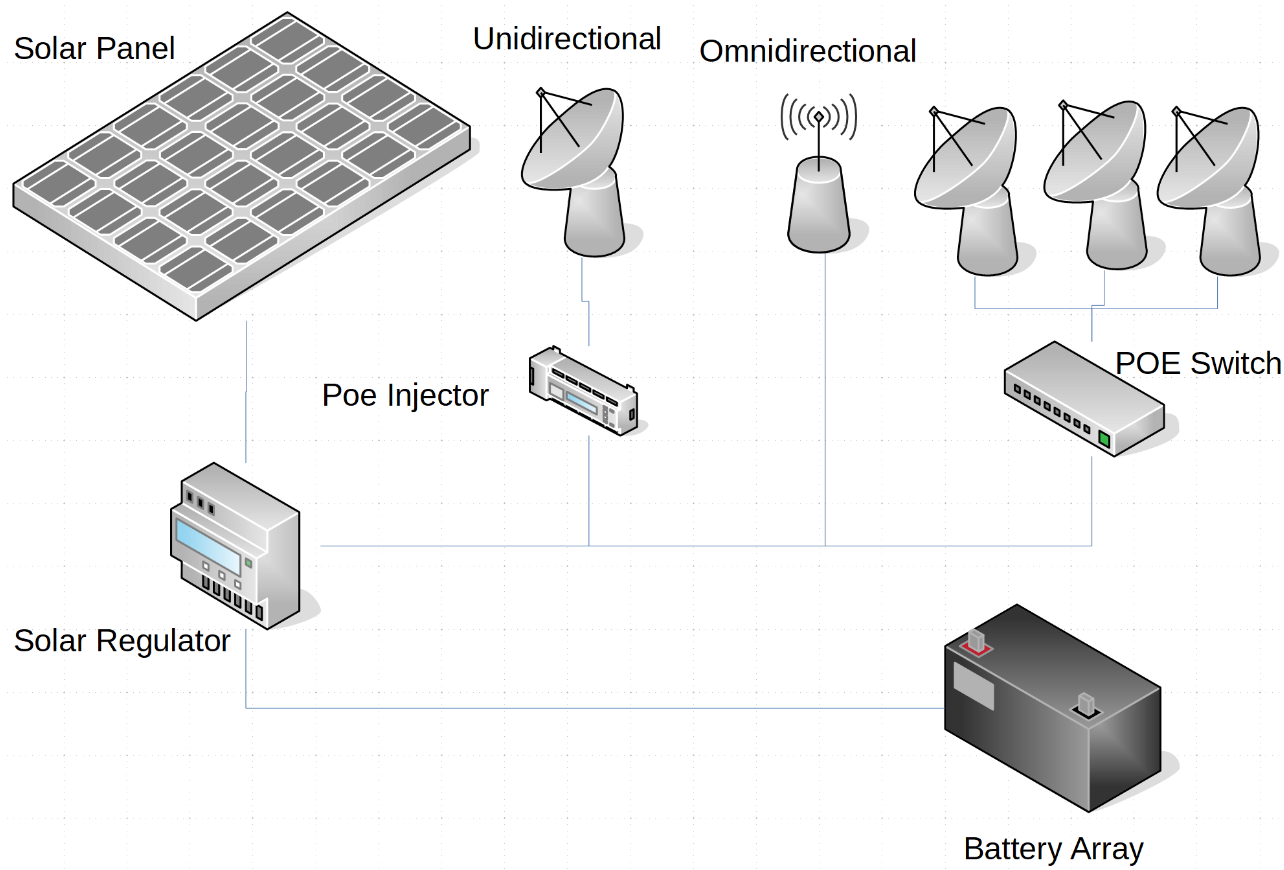

Figure 7: Example Infrastructure Node Options

Figure 7 is a diagram of a simple solar power setup with possible options for network equipment. A larger infrastructure node can be used to service a larger area, but will require a much larger battery and solar array. If using ethernet bridging on the wireless devices, then a device such as a Raspberry Pi behind the nodes can be used as the CJDNS box for that node to increase throughput. The diagrams only show power connections, not network connections. If POE injectors are utilized then any number of transmitters larger than 2 will require a switch, which will factor into the power usage. Figure 8 is an example of a POE injector setup, but utilizing a Raspberry Pi.

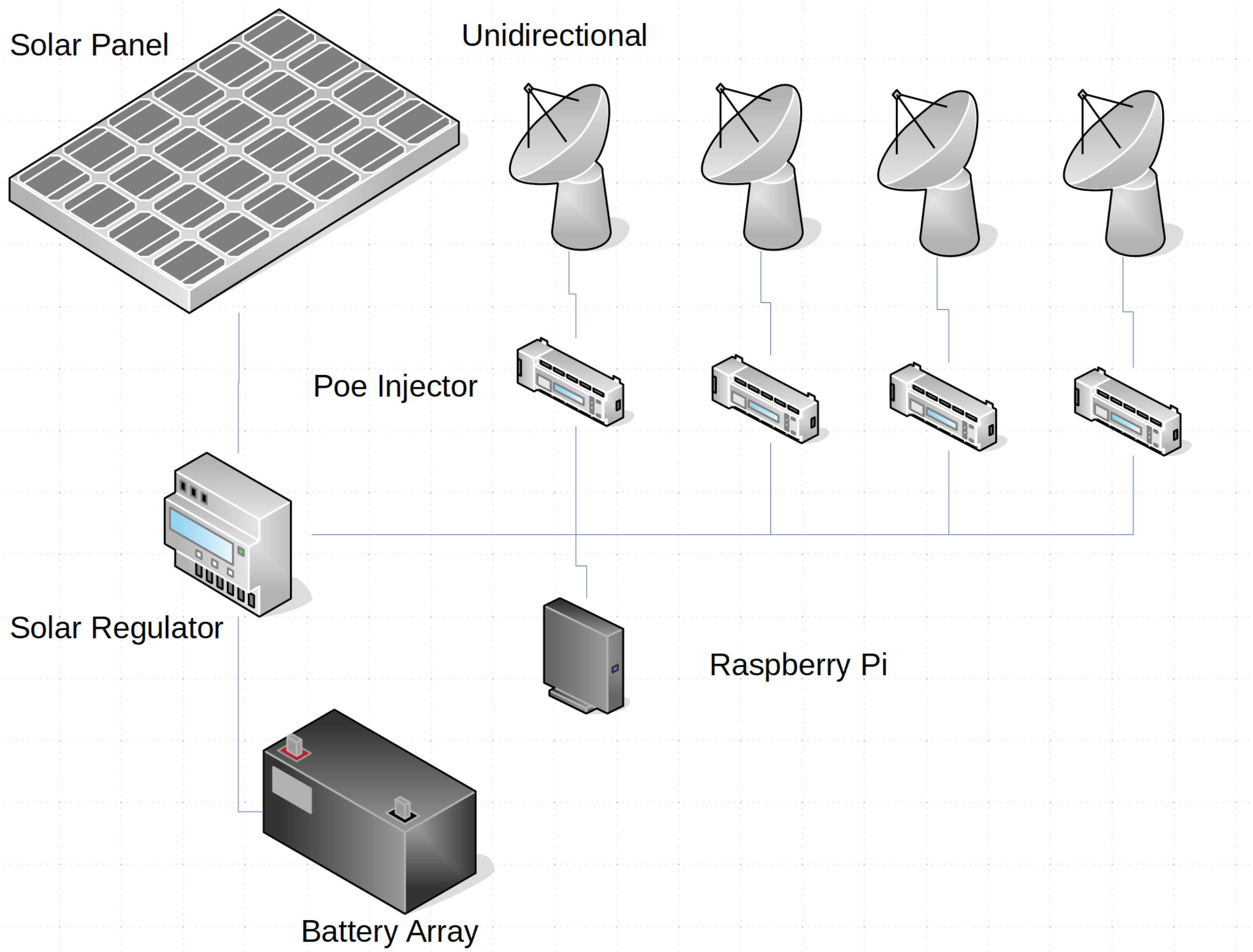

Figure 8: Example Infrastructure Node with Raspberry Pi

The following is an example case for a build design for a central mesh node.

This node requires an omnidirectional and a unidirectional transmitter, and happens to be the first link between the main area mesh network and a suburban area. This central mesh node can be pointed to from multiple locations in the suburban community, but also needs the range to reach the next nearest mesh node. Because of this, it is ideal to use 2.4GHz for the community link. For this set-up, a Raspberry Pi is used to guarantee throughput.

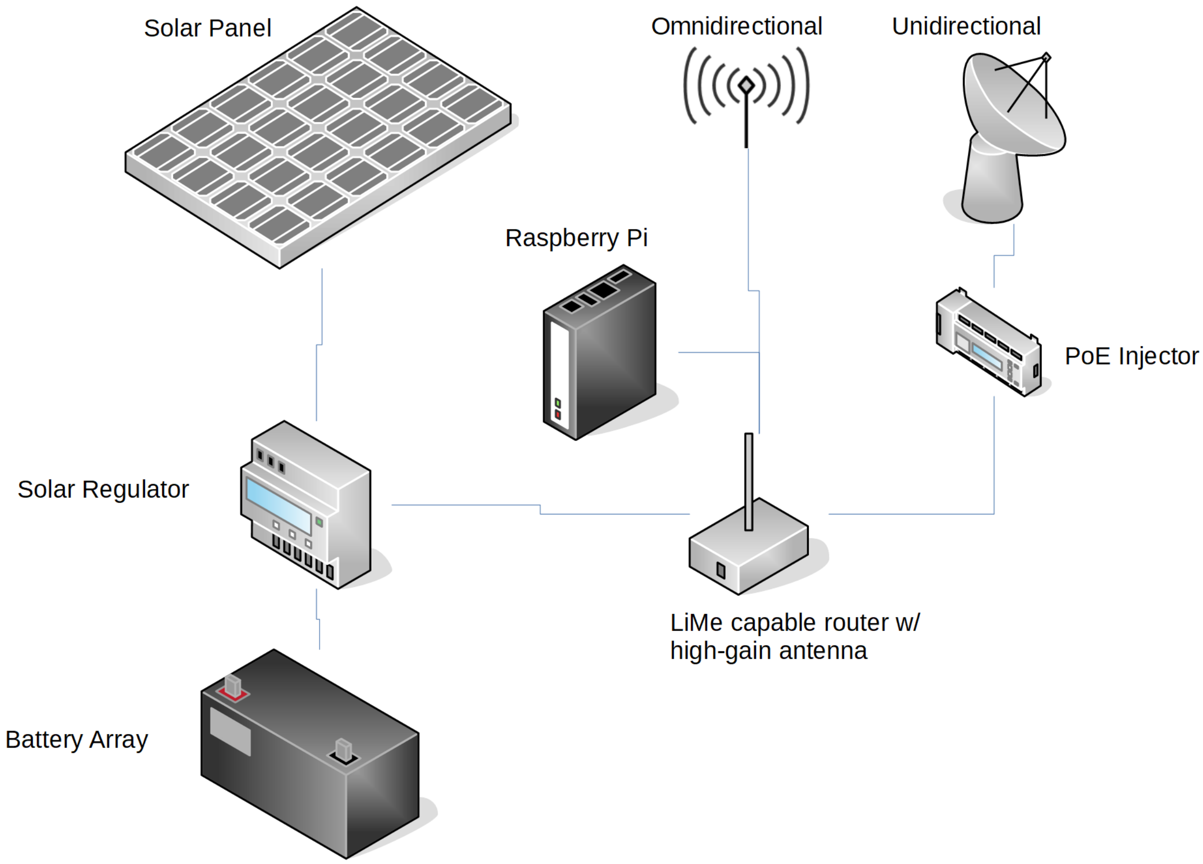

Figure 9: Example Node build for Suburban Community

Power Requirements:

Unidirectional Transmitter = .5 amps @ 24v = **1 amp @ 12v**
Router w/ Omnidirectional = 1 amp @ 12v = **1 amp @ 12v**
Raspberry Pi = 2.5 amp @ 12v = **1 amp @ 12v**
Total = **3 amps @ 12v**

Minimum battery requirement 3 amp * 24 hours = **72 amps @ 12v**

Night time amperage usage = 3 amps * 16 hours = 48 amps

Required amperage = 48 amp night time loss + 24 amp equipment usage = 72 amps

72 amps / 8 hour day = 9 amps per hour solar requirements

9 amps * 15 watts = **minimum 135 watt solar panel**

Figure 10: Example Node Power Requirements

This node doesn't have any line-of-sight, so it will have to sit on top of a building in the area. The following parts list can be used to construct the node:

- Cinder block weighted pole base
- Cinder blocks
- 8 foot pole
- Top-of-pole solar panel mount
- Weatherproof electrical Box for equipment storage
- Steel clamps for mounting equipment
- LibreMesh capable router and High-Gain Antenna
- Unidirectional Transmitter (i.e. Ubiquiti nanostations)
- Raspberry Pi
- DC PoE Injector
- 150 Watt solar panel
- 12v to 24v transformer
- 12v to 5v step downgrade
- 72 amp battery (or batteries in parallel to equal 72 amps)
- Weatherproof Battery Box
- Cat 6 cable
- Solar cables

The community infrastructure node ultimately looks like the diagram below:

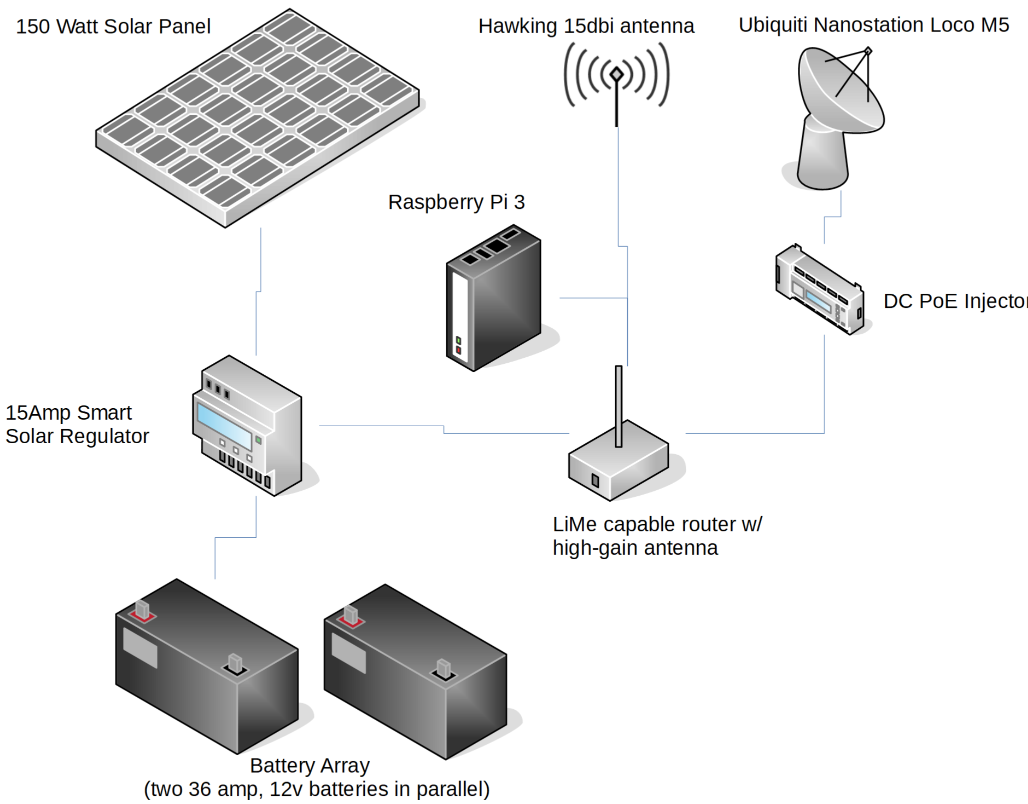

Figure 11: Example Node Build with Parts