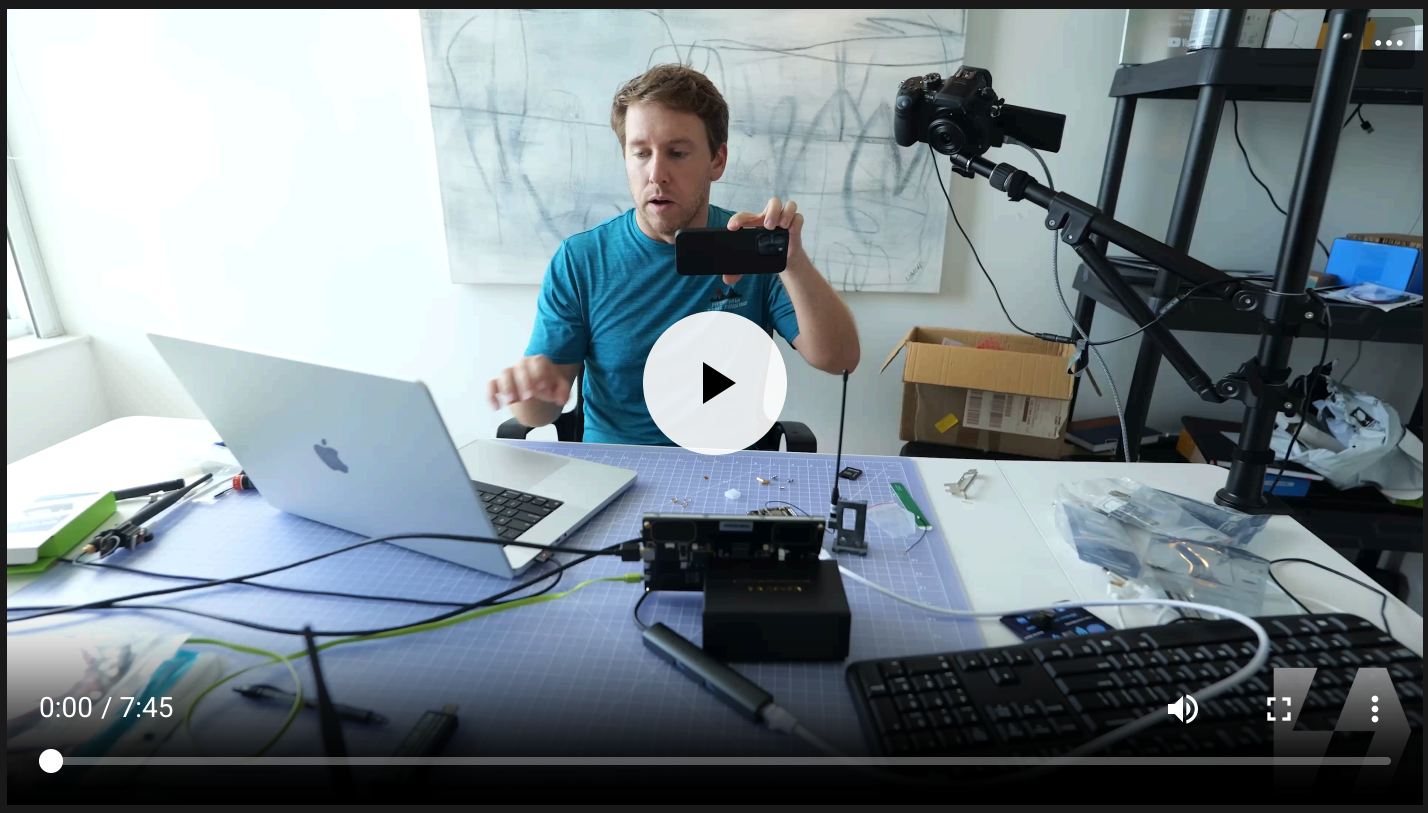
Haven – MANET IP Mesh Radio

⚠️ *Update 8/25: The broad strokes around setting this up are in place. There are a few things I'd like to add which I plan to do before this Sunday 8/31*

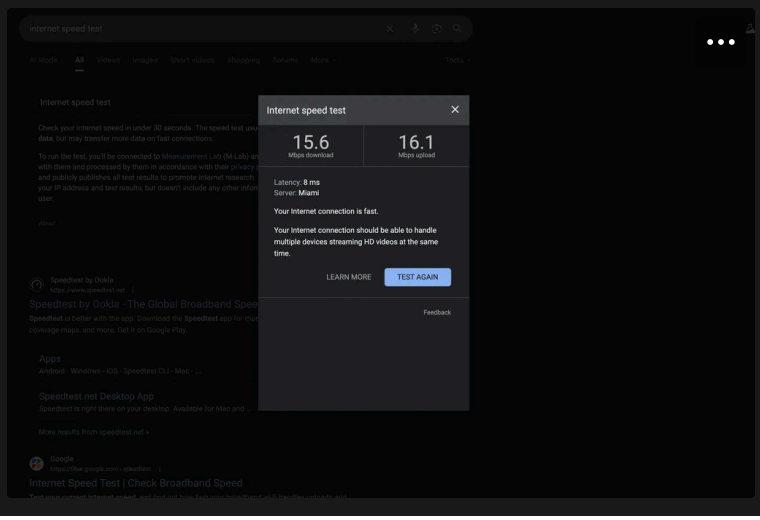
Demonstration Of What We're Looking to Build



Key Features of Haven

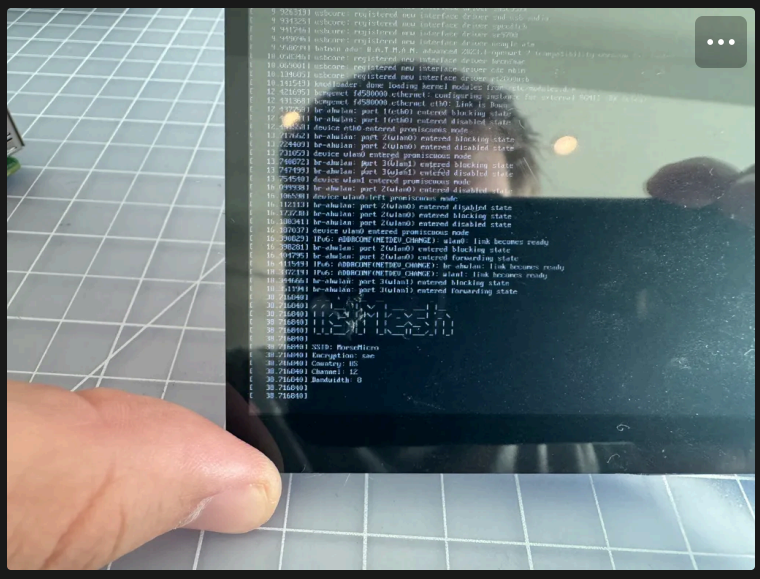
- **Proven Throughput: 15 Mbps Symmetric**

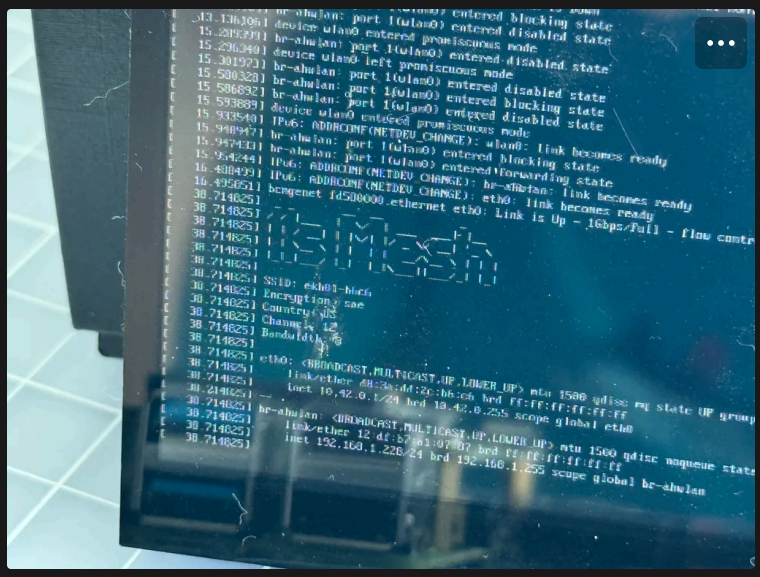
  In testing (documented in the demo video), Haven delivered a stable 15 Mbps uplink/downlink connection over HaLow mesh — more than enough for ATAK data, live video, or neighborhood-scale networking. This was with an 8Mhz channel bandwidth implementation. You will have the option to use lower or higher based on whether you want to optimize for data rate or range.



- **True Mesh Networking (802.11s)**

  Every node connects directly with every other node — no central router required. The network heals itself if a link goes down.





- **Wi-Fi HaLow (802.11ah)**

  Sub-GHz Wi-Fi standard with long range and better wall penetration than 2.4/5 GHz. Great for neighborhood or field deployments.

- **Affordable & Accessible**

  Comparable MANET radios (like the MPU5) cost $8,000+. Haven comes together with ~$106 in commodity hardware.

- **Open-Source Software**

  Built on OpenWRT and the OpenMANET project — no vendor lock-in, no subscriptions.

- **Flexible Deployment**

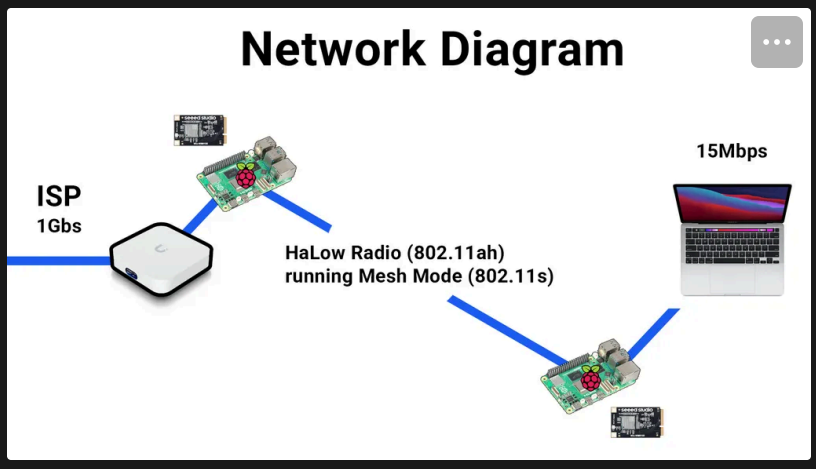
  Use indoors, outdoors, or tie into an uplink (like Starlink) for resilient backhaul.
- **Interoperability Potential**

  Can complement systems like Meshtastic or ATAK — LoRa for ultra-low-power messaging, HaLow for higher-bandwidth backhaul.

Note

To really demonstrate this you will need at least two haven nodes (the most basic mesh). In the video above I have two setups.

Network Diagram



Gear / BOM / Components

Raspberry Pi 4 Model B (8GB encouraged)

This is the motherboard computer which will run OpenWRT router operating system.

Note: This setup will not work on a Raspberry Pi 5 due to the OpenWRT build being built here.

Buy Here

Wi-Fi Halow Chip

The form factor is mini PCIe but ultimately this is a medium for the Wio-WM6108 developed by MorseMicro.

Full chip spec

Wi-Fi HaLow Modulation and Coding Scheme

MCS index	Modulation scheme	Coding rate	PHY rate (kbps) per BW			
			1 MHz	2 MHz	4 MHz	8 MHz
10	BPSK	1/2 x 2	167		N/A	
0	BPSK	1/2	333	722	1500	3250
1	QPSK	1/2	667	1444	3000	6500
2	QPSK	3/4	1000	2167	4500	9750
3	16-QAM	1/2	1333	2889	6000	13000
4	16-QAM	3/4	2000	4333	9000	19500
5	64-QAM	2/3	2667	5778	12000	26000
6	64-QAM	3/4	3000	6500	13500	29250
7	64-QAM	5/6	3333	7222	15000	32500

Buy Here

Mini PCIe Adapter Hat for the Raspberry Pi

Makes use of the Pi's GPIO pins to fix the HaLow chip onto the pi.

Buy Here

Antenna

We need an antenna that can handle sub gigahertz ranges.

- **United States (FCC): 902–928 MHz**
- **Europe (ETSI): 863–868 MHz** and **915–921 MHz**

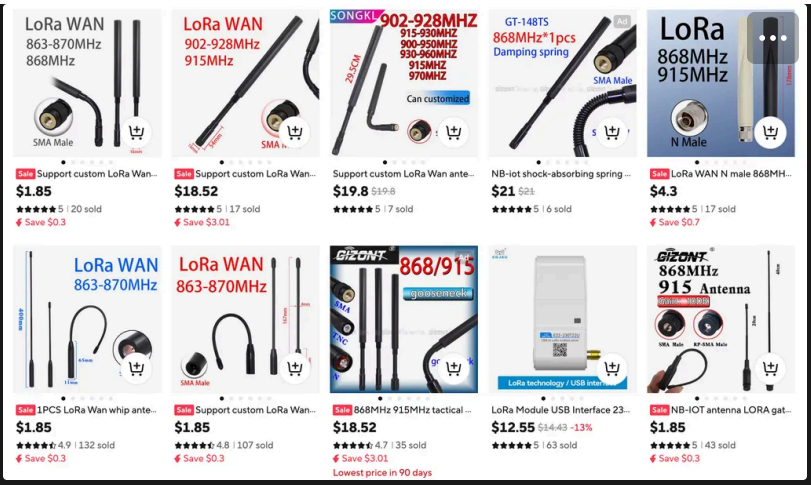
Meshtastic/LoRa antennas will work (it's what I use here).

915Mhz Whip Antenna

Buy Here

There may be other better more appropriate antennas. I'm thinking about getting some of these more rugged goose neck antennas. AliExpress has some interesting gear for prototyping.

https://www.aliexpress.us/w/wholesale-LoRa-Wan-antenna.html?spm=a2g0o.detail.search.0



U.FL to SMA Connectors

You may want some spare or extended u.fl to sma connectors.

Buy Here

Wi-Fi USB Adapter

You'll likely want to connect to your Haven via Wi-Fi instead of just ethernet. I have not been able to get this to work but according to the OPENMANET docs. It should work with this USB wi-fi adapter. The Raspberry Pi has a wifi chip set onboard but it competes with the halow card and therefore cannot be used with halow in tandem. One option is to simply add a wifi usb adpater (link below). Other options include using ethernet or ethernet to a traditional router like Unifi, tp-link, or linksys. But the most optimal setup would be to use the wifi usb adapter. (I'll be testing this shortly)

Buy Here

Accessories

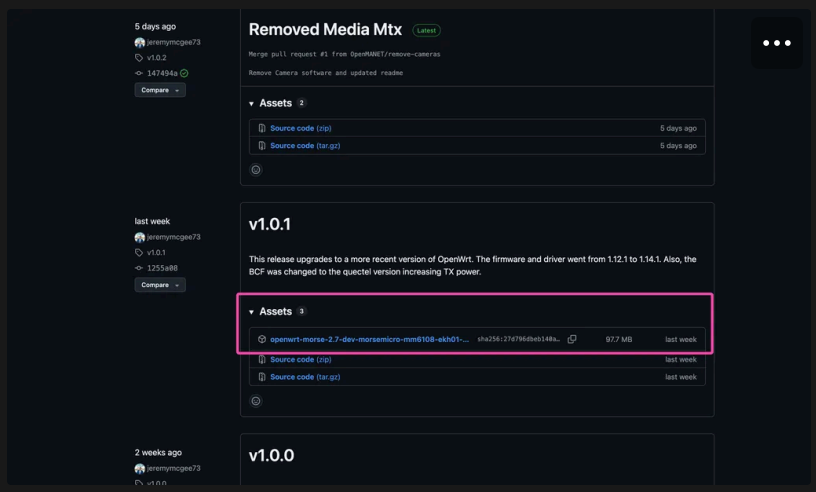
- Ethernet cable
- Raspberry Pi power supply or USB-C power cable
- MicroSD card (>32GB)
- Ethernet to USB-C (you will need to run ethernet into your computer somehow for configuring the node, on my mac I needed a converter to establish this connection)

Assembly

Both nodes will run the same firmware image but they will be configured slightly differently. OpenWRT has a wizard that will allow you to try different routing structures but I will show what I used for my example above. Even better setup may exist for your use case.

Acquire the OpenWRT image from the following github repo.

https://github.com/OpenMANET/openwrt/releases



Or you can download it here.

openwrt-morse-2.7-dev-morsemicro-mm6108-ekh01-squashfs-sysupgrade-addbcf.img.gz 97.7 MB

The guy who built this image put in a lot of elbow grease to do two things.

1. enable 802.11s
2. overclocked the tx Power from 21 dbm to 27 dbm (which should increase the range)

If you want to say thanks you can.

Buy him a coffee

No need to unzip/unpack.

This file is an OS system image that will be flashed to your Raspberry Pi. It's a fork of OpenWRT developed by MorseMicro.

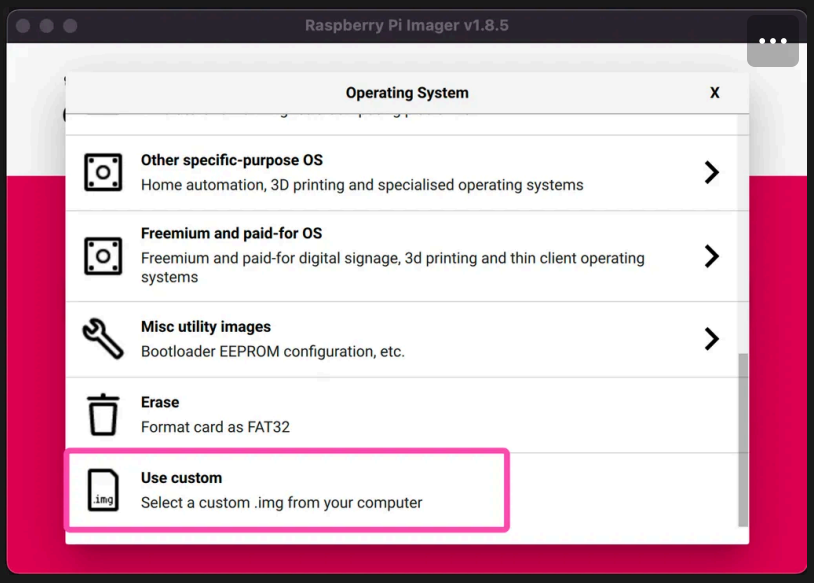
Note: You can generate your own system image by following the MorseMicro build instructions.

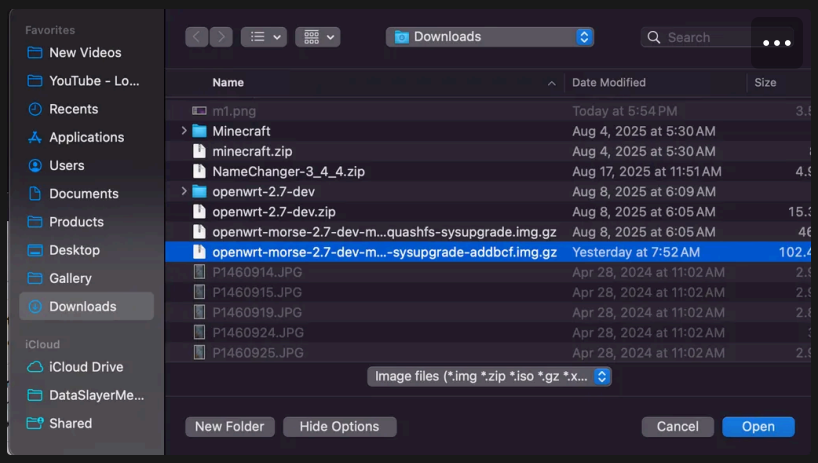
https://github.com/MorseMicro/openwrt

I've messed around with this and do note that there are a number of configurations settings that will be enabled to support the Raspberry Pi architecture, halow and 802.11s which, frankly, is a massive pain in the ass.

Another alternative is to use the guide and tutorials produced by Seeed Studio (I've run these and they work) but they don't have 802.11s (meshing).

Once the image is downloaded use Raspberry Pi imager to flash it to a micro sd card.





No need to add configs or wifi settings.

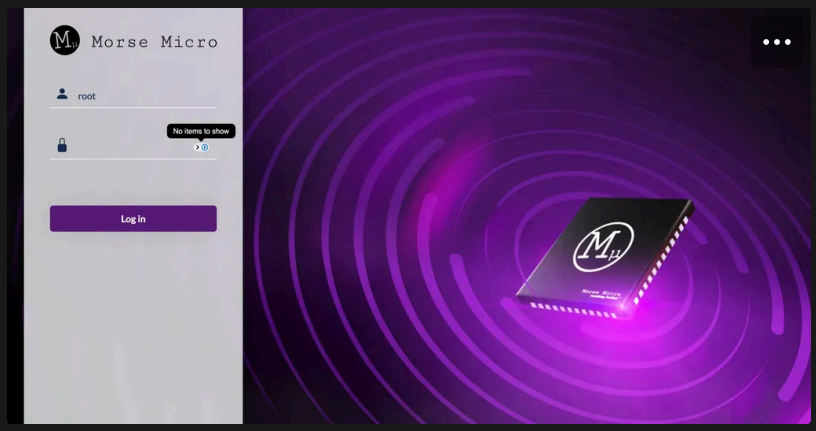
Once it's ready, slide the sd card into the pi and boot it up.

For any node type you can boot up the Raspberry Pi and connect it to your computer. You may need to turn off wi-fi.

Visit this ip in your browser

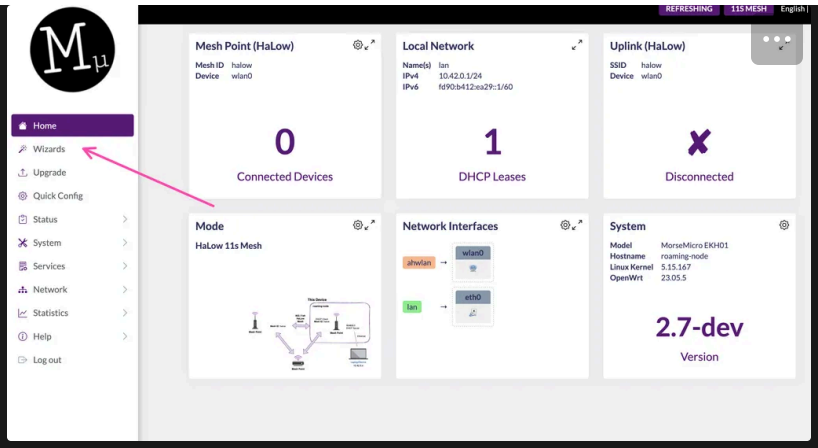
```javascript
10.42.0.1
```

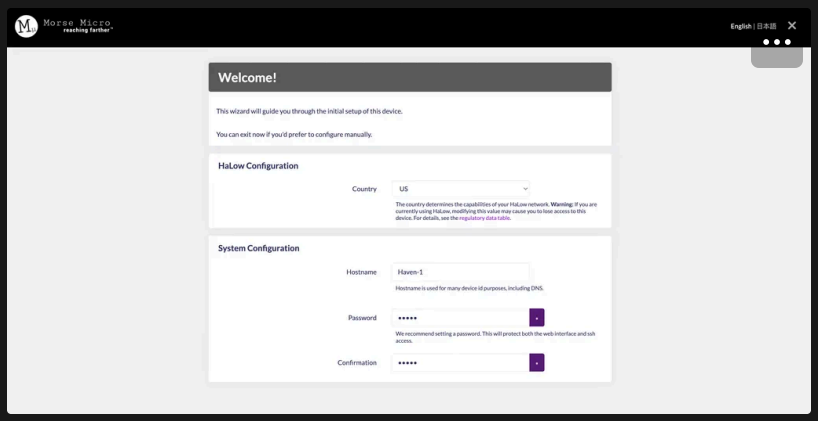
Node 1 (Roaming Client)



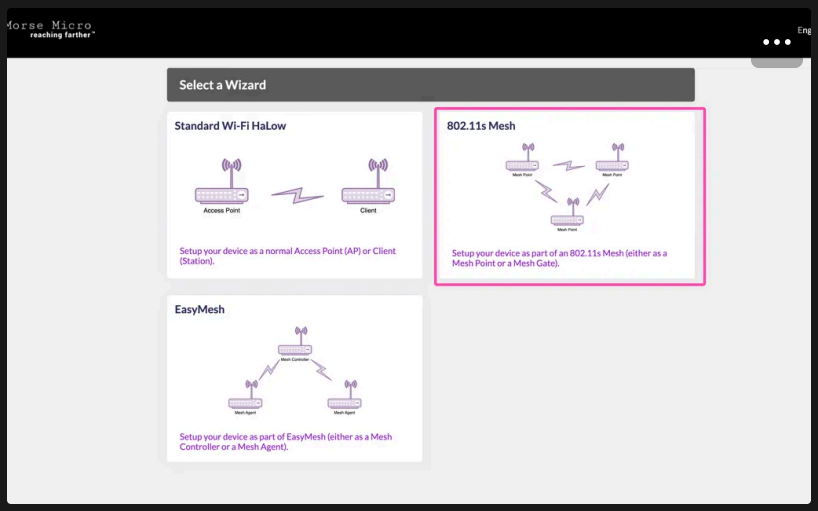
You should be able to login with **no password entered**.

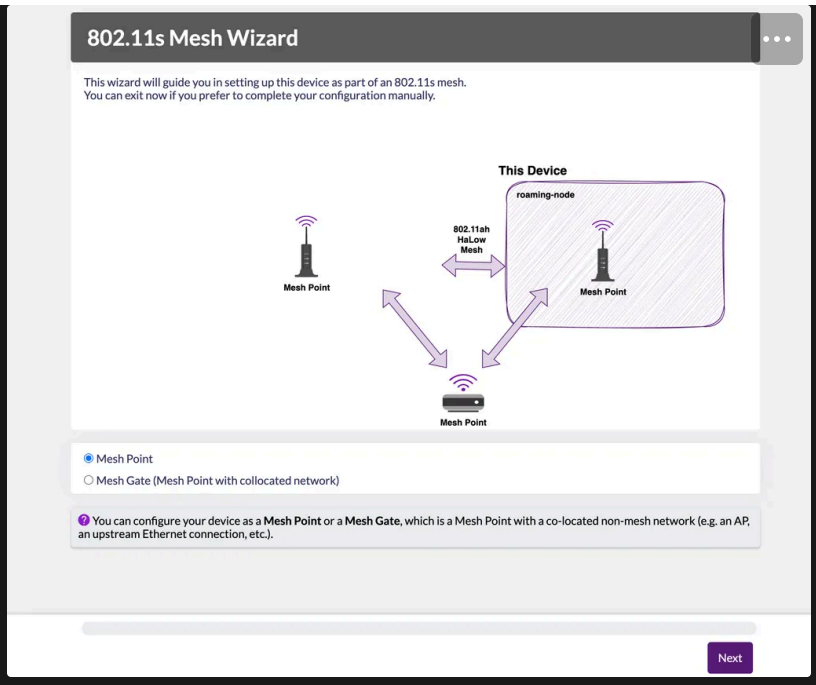
It may shoot you right to the Wizard but if not you can navigate there manually.

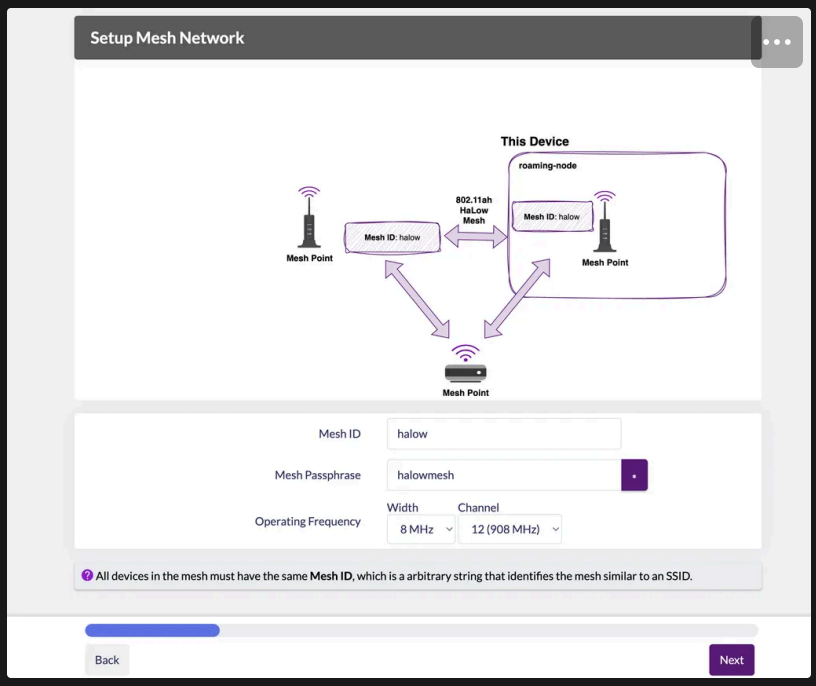




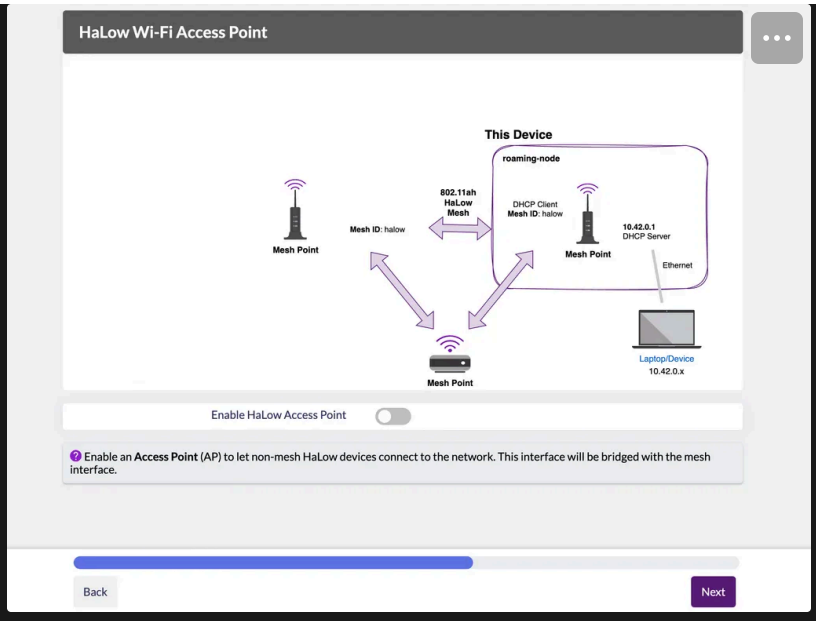
This password is for logging into the UI.



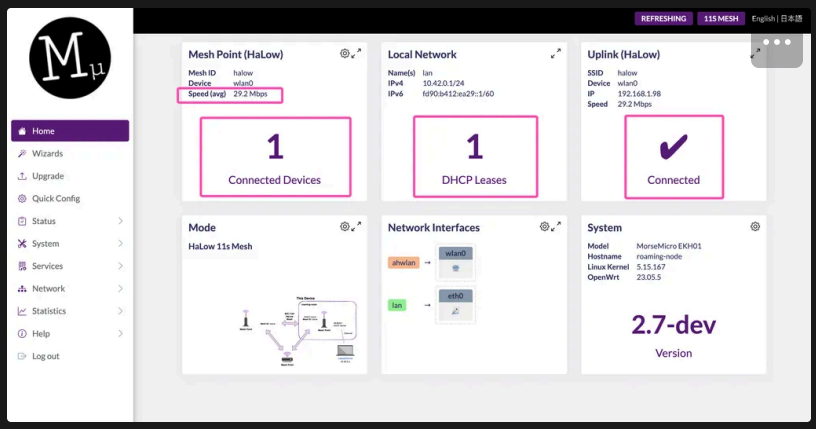




Next screen set as “bridge”



When your other node is setup you can come back to the dashboard and you'll see some output like this.



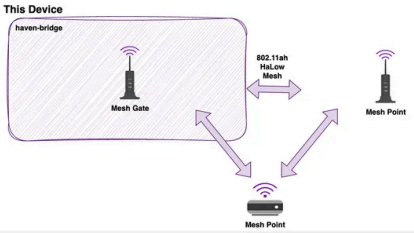
Node 2 (Uplink)

For the node you want to connect to an uplink (ISP/fiber/starlink) follow these instructions for the setup.

Mostly the same setup as above but when you get to the Wizard select the following.

802.11s Mesh Wizard

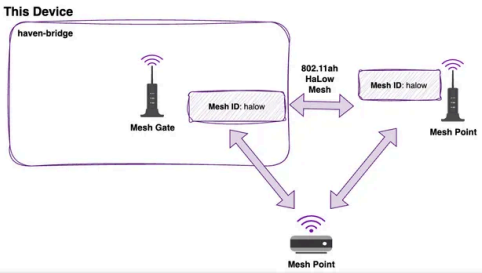
This wizard will guide you in setting up this device as part of an 802.11s mesh.
You can exit now if you prefer to complete your configuration manually.



- ○ Mesh Point
- ◉ Mesh Gate (Mesh Point with collocated network)

An 802.11s **Mesh Gate** provides both a Mesh Point and a co-located non-mesh network (e.g. an AP, an upstream Ethernet connection, etc.). It broadcasts mesh gate announcements to help align the mesh nodes, making it easier for traffic to reach the non-mesh network.

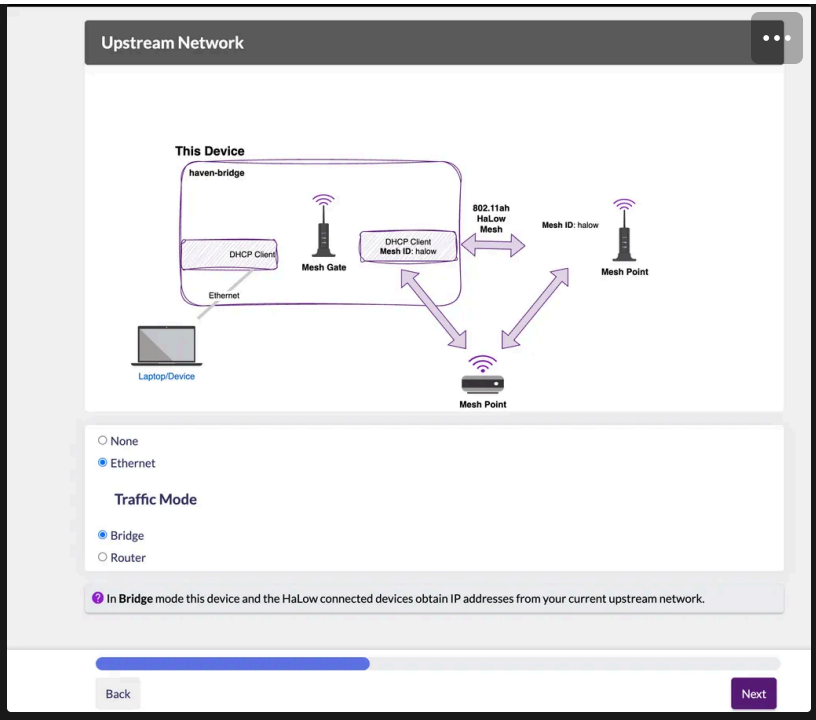
Setup Mesh Network

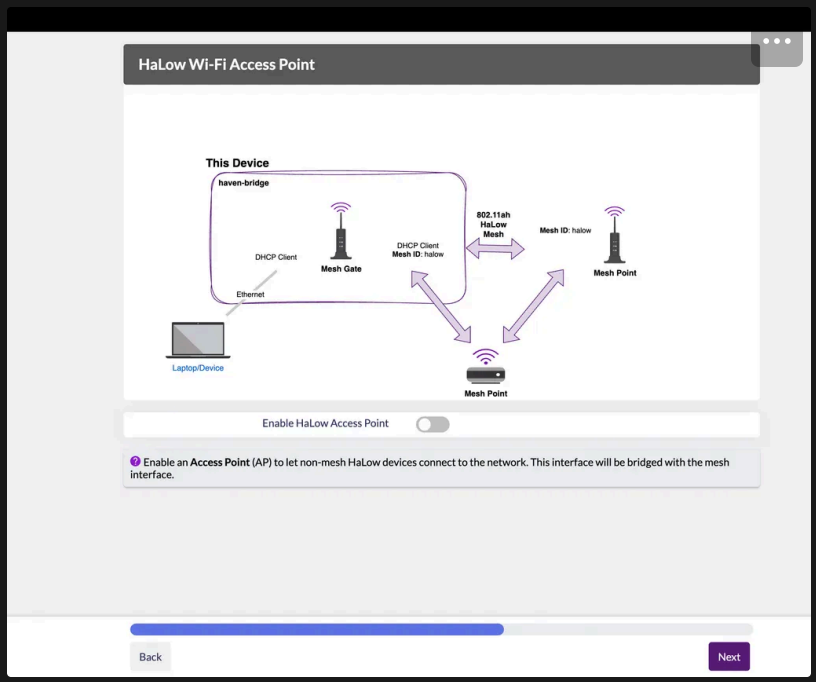


Mesh ID	halow	
Mesh Passphrase	halowmesh	•
Operating Frequency	Width: 8 MHz	Channel: 12 (908 MHz)

All devices in the mesh must have the same **Mesh ID**, which is a arbitrary string that identifies the mesh similar to an SSID.

Back | Next





Finally, connect this node to your uplink via ethernet.

With this setup you could add new roaming nodes. All ip networking is bridged.

Performance

How to run a performance test.

How to log into your nodes.

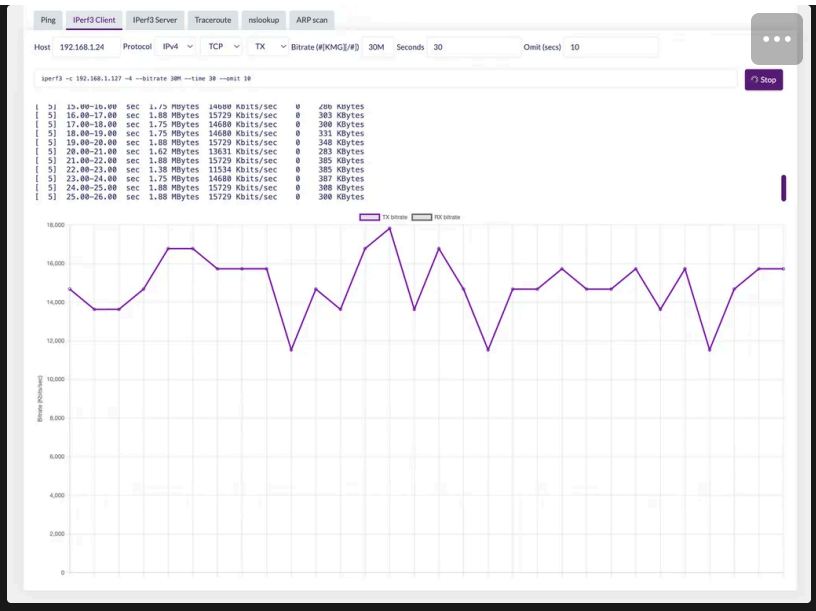
Below is a series of iPerf test I ran to determine throughputs at various channel widths and frequencies. They align with the expected theoretical MAX PHY rates which are determined by the modulation technique in use (802.11ah)

Performance By Configuration

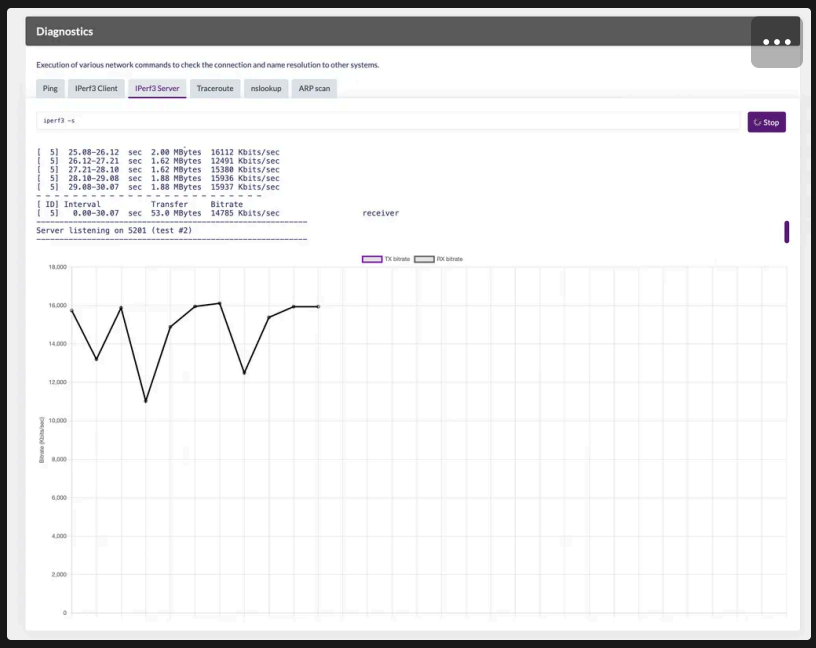
Channel Width (MHz)	Channel Frequency (MHz)	Throughput (Mbps)	Distance (meters)

Running iPerf between the two nodes.

Node 1



Node 2



> iPerf3 test shows the two HaLow nodes sustaining about **15 Mbps symmetric throughput** with stable performance and minimal loss. That's more than enough for voice, video, and ATAK data, putting Haven well above typical LoRa/Meshtastic kbps links.

In this example we don't talk about standard 2.4/5GHZ wi-fi (which you'd likely use). I don't believe the on board wifi chip can be used with this setup, however you should be able to get it going with an inexpensive USB wifi hub like the

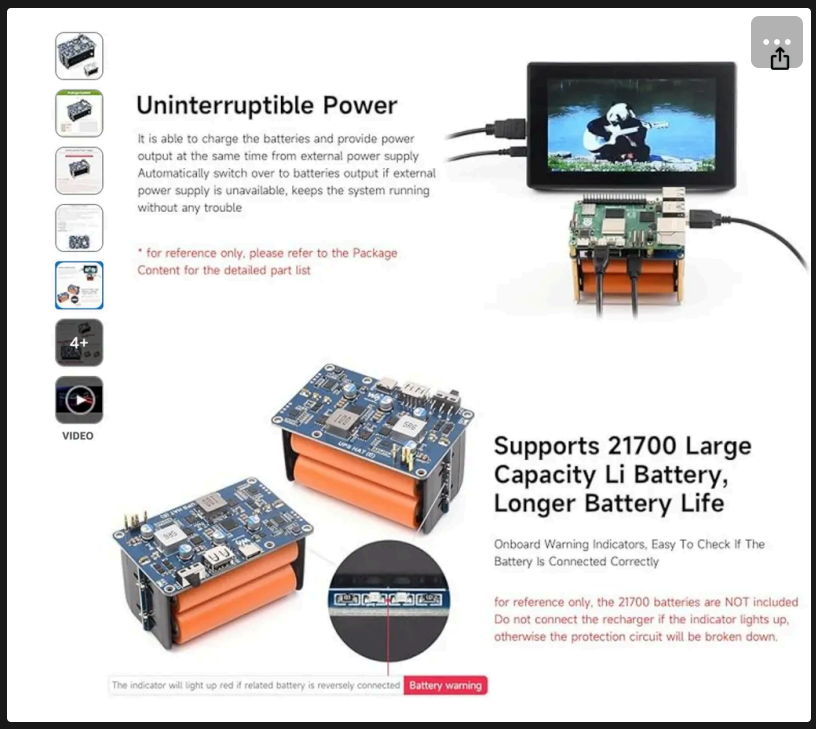
I'll play around with this and add details but I'm 99% confident it's doable which would be great for pulling ATAK into the mix.

Future

Would love to test additional settings like...

1. Range tests
2. Channel widths

3. Frequency
4. Antenna form factors
5. Expanded mesh
6. 3D Print STL files for this beast
7. Cool power options like this…



Data Slayer & Parallel Community