Dual Radio Dynamic Mesh User Guide

Introduction

In this guide, we discuss the Mesh Rider dual-radio Dynamic Mesh configuration. This configuration is designed for high resiliency, anti-jamming UAV to GCS or other robotics links.

In this guide, we will discuss two configurations. A high-level system diagram of the dual-radio redundant link is shown in Fig. 1. In this configuration, packets are sent redundantly over both radio links, so that the effective link is impervious to interruptions in either link.

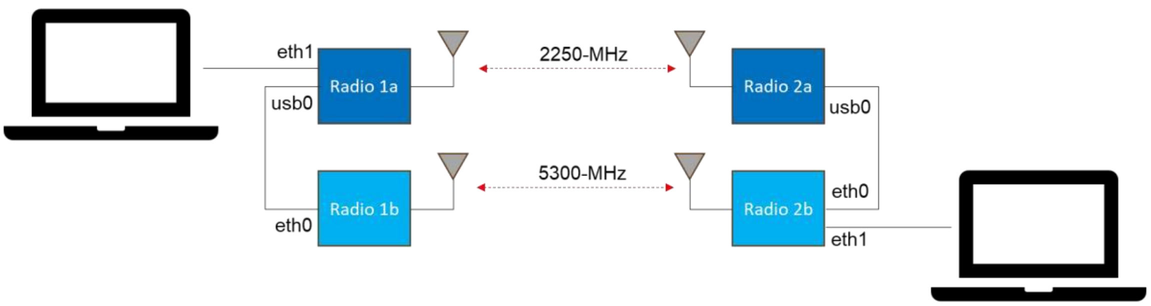


Fig. 1 Dual-radio redundant link configuration

The second configuration is a relay mode shown in Fig. 2.

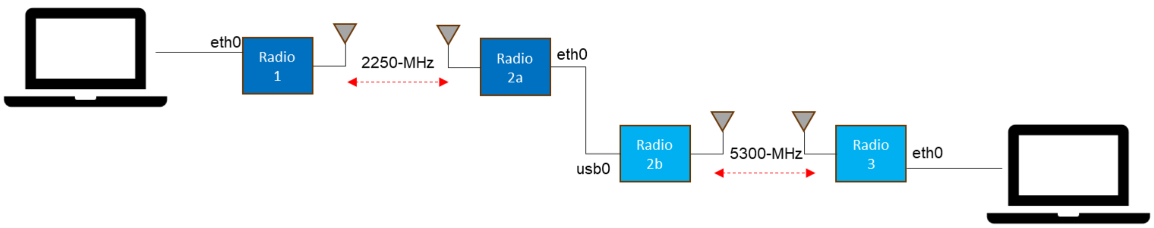


Fig. 2 Dual-radio relay link configuration

Hardware Setup

Redundant Link Configuration

Referring to Fig. 1, we are using RM-2025-62O3 and RM-5200-92O3 as Radios 1a/2a and 1b/2b respectively. They are deliberately set up to ensure an even link across both radio links.

Software-wise, the connections between the radios and the connections to the PCs are interchangeable as long as the radios are configured properly. However, as the USB interfaces (usb0 and eth0) provide the highest throughput, we recommend using them for the radio-to-radio connection. The cable harnesses provided with the dual-radio evaluation kits only include USB-host to USB-device cables. The setup of a single side of the link is shown in Fig. 3. Antenna connections are not shown.

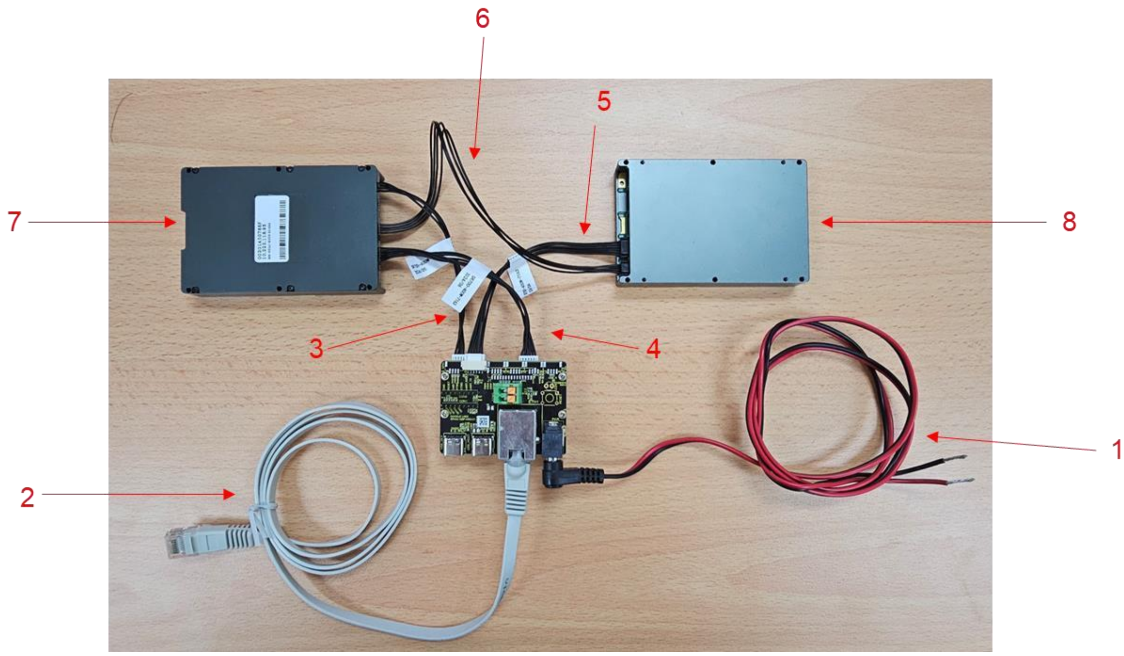


Fig. 3 Setup on one side of the link

1. External Power, 6-24 V, 30-W
2. Ethernet connection to external host
3. Ethernet connection to Radio 1a (or 2b)
4. Power connection to Radio 1a (or 2b)
5. Power connection to Radio 1b (or 2a)
6. USB connection between Radio 1a and 1b (or 2a and 2b)
7. Location of antenna connectors (mmcx) on Radio 1a (or 2b)
8. Location of antenna connectors (mmcx) on Radio 1b (or 2a)

A closer look at how the two radios are connected to one another is shown in Fig. 4.

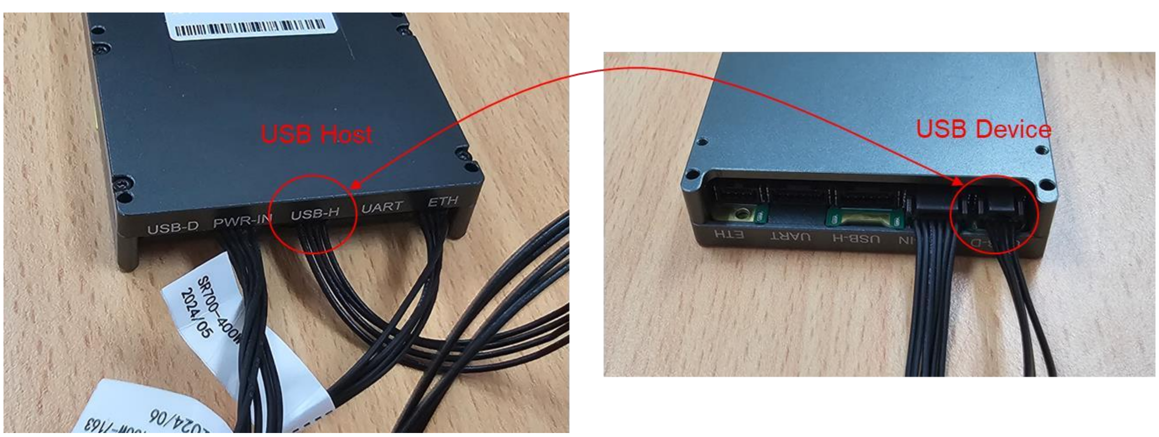


Fig. 4 Interconnection between two radios

A closer look at the connections on the EVK board is shown in Fig. 5. The power connectors are interchangeable.

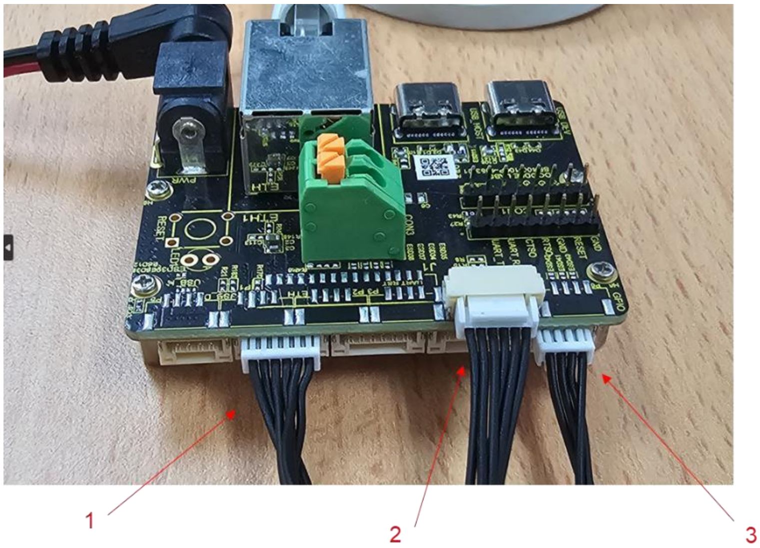


Fig. 5 EVK connections

1. Power
2. Power
3. Ethernet

Relay Configuration

In the relay configuration (Fig. 2), Radios 1 and 3 do not require a special hardware setup. They can be set up in the same way that you would normally set up a single radio. The only caveat is that if you use the default software configuration, then USB-H and USB-D will be used for mesh extension and only eth1 will be available for communication with an end host. Radios 2a and 2b can be set up following the guidelines above.

Software Configuration

Dual Radio Dynamic Mesh Evaluation Kits are set up for you in the correct configuration as long as you use the hardware setup suggested above. This section discusses how you can apply the configuration yourself.

An important note is that you need to access each radio individually to configure it or to monitor its status.

License upload

Before applying any configuration, we recommend first installing your Dynamic Mesh and *Sense* licenses. You can install the licenses through the web GUI. From here you can also check the status of your licenses. Dual Radio Dynamic Mesh EVKs come with these licenses pre-installed.

- For Dynamic Mesh, navigate to the *Network Configuration → Dynamic Mesh* menu.
- For Sense, navigate to the *System → License Manager* menu.

You can also upload licenses manually by

- Copying Sense licenses to the /opt/licenses folder in the radio with the name sense.license.
- Copying Dynamic Mesh licenses to the /opt/ folder in the radio with the name dynamic_mesh.license.

After uploading your licenses, navigate to the Simple Configuration menu.

Simple Configuration

Dual Radio Redundant Link Configuration

The recommended way to configure the radios is through the Simple Configuration menu. Fig. 6 shows the recommended setup on one of the radios. For other radios, use the appropriate operating band and channel.

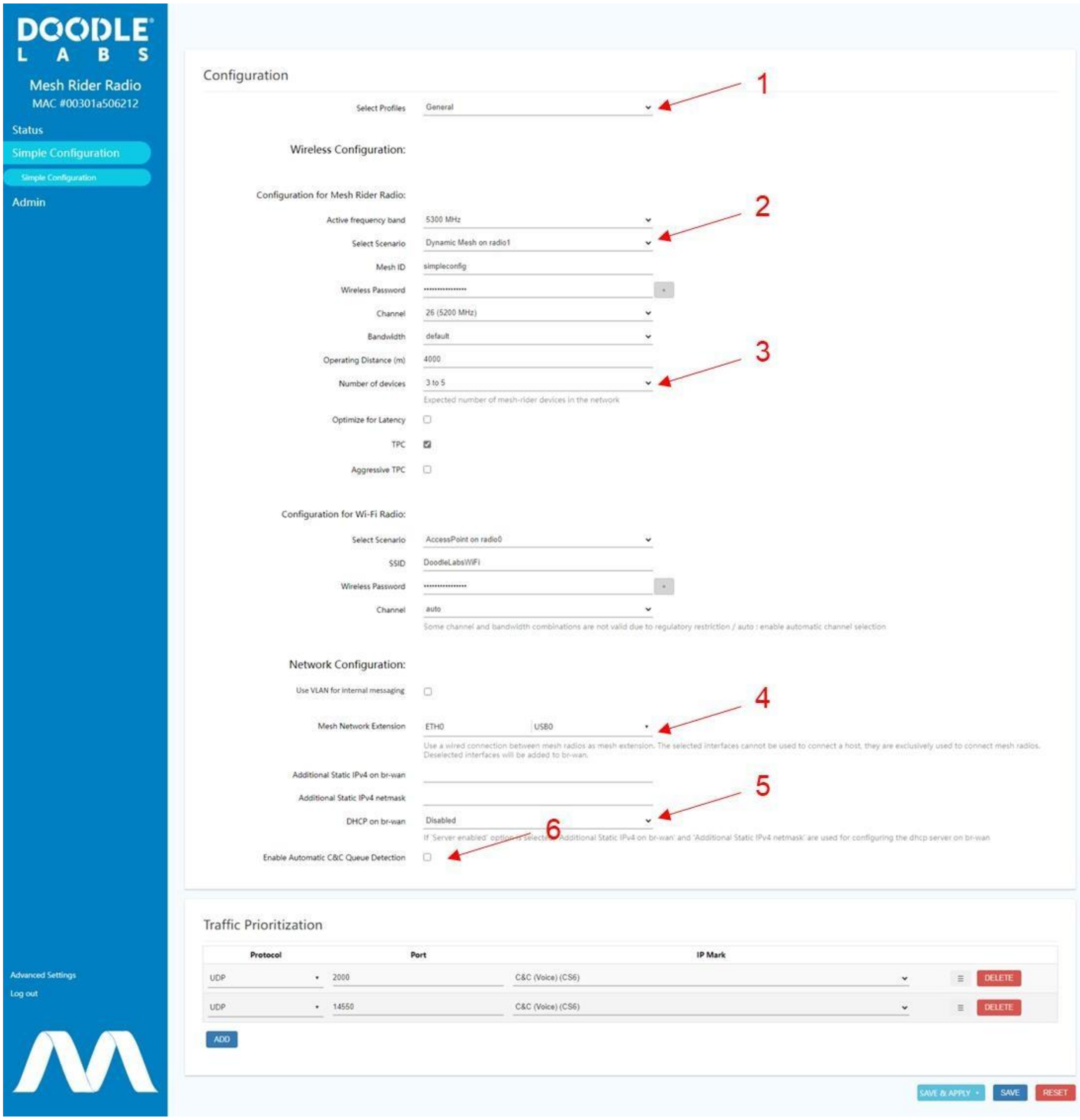


Fig. 6 Dynamic Mesh configuration for dual-radio system

1. Use the General profile
2. Choose Dynamic Mesh
3. Select 3 to 5 devices

4. Choose ETH0 and USB0 as the “Mesh Network Extension” interfaces. Note that ETH0 is the USB device interface. Do not choose all interfaces as “Mesh Network Extension” interfaces, as you cannot access your radio normally over these interfaces.
5. (optional) Disable “DHCP on br-wan”.
6. (optional) Disable “Automatic C&C Queue Detection”.

Dual Radio Relay Configuration

The configuration for Radios 2a and 2b is the same as already discussed above. On radios 1 and 3, the only difference is in point 4 in the section above. For radios 1 and 3, we don’t need to select any interfaces as “Mesh Network Extension” interfaces.

Dual Radio Dynamic Mesh Usage

From the perspective of the hosts at the two ends of the link, the Dual Radio Dynamic Mesh hardware acts like an Ethernet switch and passes all IP traffic just like in the single-radio use case.

Dynamic Mesh monitoring

After applying the Dynamic Mesh configuration in the Simple Config page, you can get detailed information about Dynamic Mesh and perform further configuration in the *Network Configuration → Dynamic Mesh* menu.

Fig. 7 shows a screenshot of the *Routes* tab. The *Routes* tab shows us how data is routed to each Dynamic Mesh node. The first radio in the table is connected over the USB-D to USB-H cable. Each of the other radios are redundantly connected to this node.

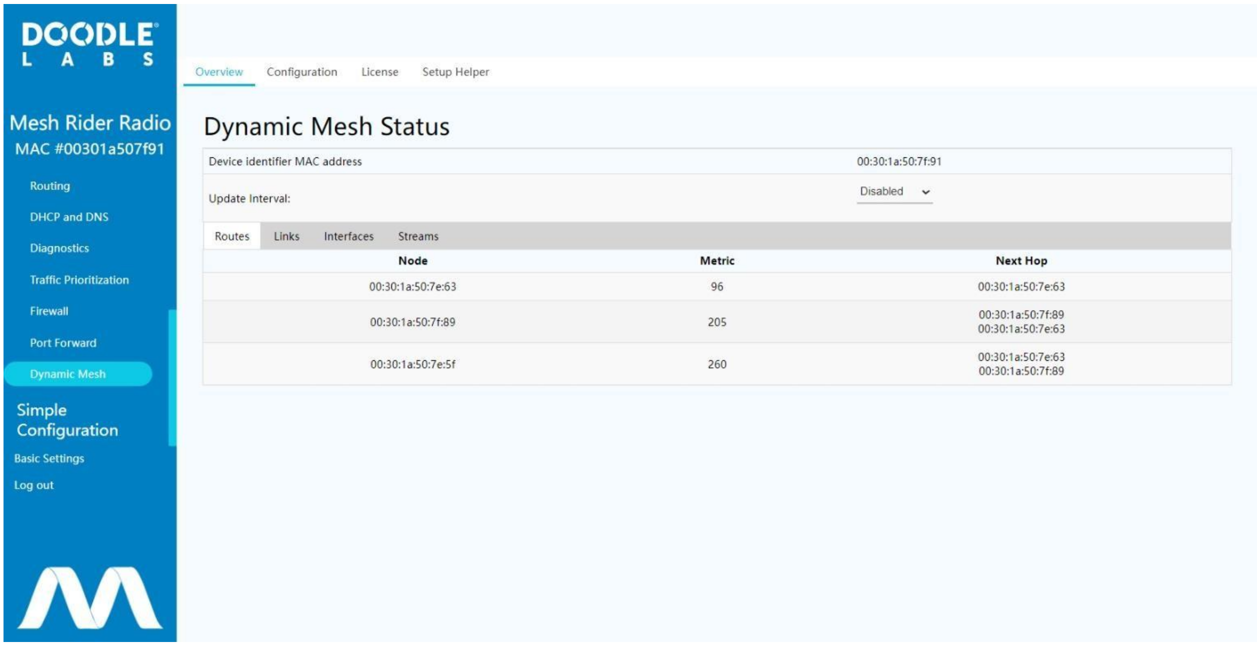


Fig. 7 Dynamic Mesh Routes

Fig. 8 shows a screenshot of the *Links* tab. The *Links* tab shows us the nodes that this node is directly connected to along with link information. Nodes that are not directly reachable are not shown in this table.

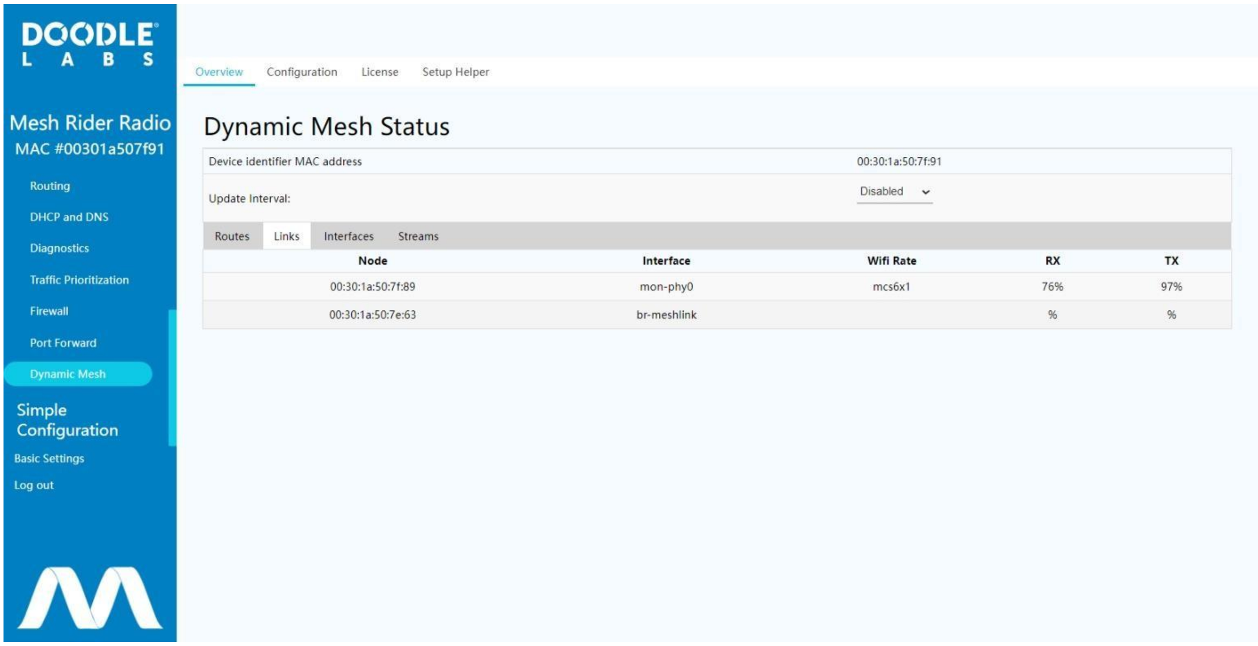


Fig. 8 Dynamic Mesh Links

The *Interfaces* tab shows basic information about device interfaces. The *Streams* tab shows advanced debugging information and will not be discussed.

The **Configuration tab** is shown in Fig. 9. The only configuration setting which you may want to change is the *Unicast Routing* setting. *Multipath* mode tells Dynamic Mesh to route packets redundantly while *Shortest Path* tells Dynamic Mesh to only route packets over the best path. If *Shortest Path* routing is used, there will be a momentary link loss (a few seconds) when the optimal path changes. The *Hive Worker* section is not in use currently.

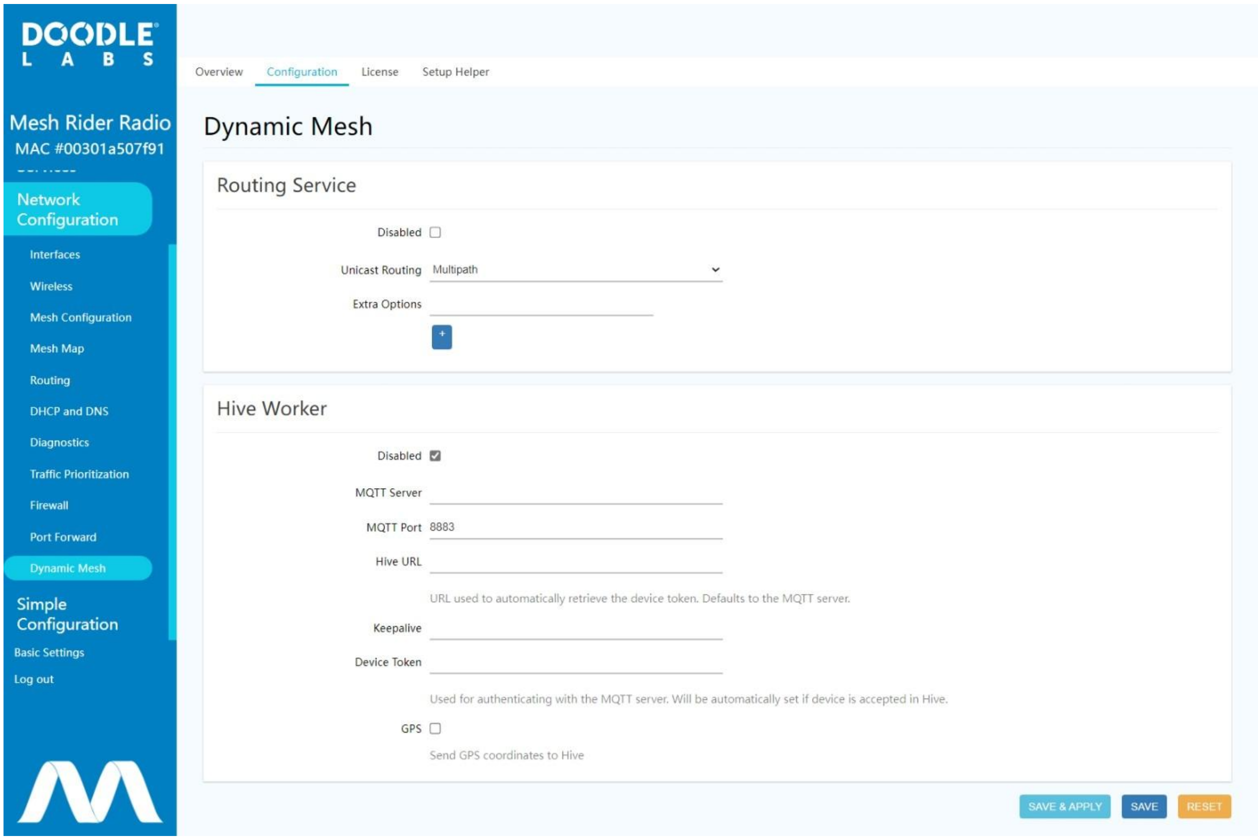


Fig. 9 Dynamic Mesh Configuration

The *License* tab is where you can upload your *Dynamic Mesh* license. You do not need to register your device. The *Setup Helper* tab can also be ignored, and you should instead set *Dynamic Mesh* up in the *Simple Config* menu.

CLI tools

If you SSH into the unit, you can run the following commands relating to *Dynamic Mesh*.

- dynamic_mesh link shows us direct link information.

```
root@smartradio-301a3bdf6f:~# dynamic_mesh link
     remote_iface       remote_node type stale rx/tx:    mcs0x1    mcs2x1    mcs4x1    mcs6x1
00:30:1a:51:07:3e 00:30:1a:3c:03:1d wifi     1      :  98%/ 97%  59%/  0%   1%/  0%  83%/ 40%
00:30:1a:3b:51:8e 00:30:1a:3b:51:8d  eth     1      :
```

Field Explanations:

remote_iface – The local interface MAC address being used to talk to the peer.

remote_node: The peer's MAC address

Type: Link Type – wifi for wireless, eth for wired

Stale: Whether the link info is stale (1= stale, 0 = fresh). Stale indicates the entry has not been updated this polling cycle

Rx/tx: Probability of success for rx/tx for the given MCS rate

- dynamic_mesh route shows us routing information.

```
root@smartradio-301a3bdf6f:~# dynamic_mesh route
00:30:1a:3b:51:8d metric 96 seqno: 8 p: 100% stale: 1 relay: 0 countdown: 0
        via 00:30:1a:3b:51:8d (255) iface: br-meshlink
        relay 00:30:1a:3b:51:8d metric 0 seqno: 8 p: 100%
        relay 00:30:1a:3c:03:1d metric 280 seqno: 7 p: 98%

00:30:1a:3b:51:99 metric 321 seqno: 94 p: 96% stale: 1 relay: 0 countdown: 0
        via 00:30:1a:3c:03:1d (255) iface: mon-phy1 rate: mcs6x1
        via 00:30:1a:3b:51:8d (127) iface: br-meshlink
        relay 00:30:1a:3b:51:8d metric 276 seqno: 94 p: 98%
        relay 00:30:1a:3b:51:99 metric 0 seqno: 188 p: 100%
        relay 00:30:1a:3c:03:1d metric 96 seqno: 94 p: 100%

00:30:1a:3c:03:1d metric 225 seqno: 56 p: 96% stale: 1 relay: 0 countdown: 0
        via 00:30:1a:3c:03:1d (255) iface: mon-phy1 rate: mcs6x1
        via 00:30:1a:3b:51:8d (127) iface: br-meshlink
        relay 00:30:1a:3b:51:8d metric 301 seqno: 55 p: 97%
        relay 00:30:1a:3c:03:1d metric 0 seqno: 56 p: 100%
```

- dynamic_mesh license <license file> shows us detailed license information.

```
root@smartradio-301a3bdf6f:~# dynamic_mesh license /etc/dynamic_mesh.license
Valid license for wlan0
```

Sense Configuration

To enable the licensed **Sense** feature on your system, each individual radio must be loaded with its own valid license. When Sense is enabled on *both* links, you'll get the highest resiliency and best performance in challenging RF environments.

The process for configuring Sense is the same as described in the **Advanced Sense Debugging Guide** provided by your Sales Representative, with one important difference:

Each link must use a different network port for Sense communication.
Sense uses this port to send data to its peer radio. If both links share the same port number, their data traffic can interfere with each other, causing conflicts and reduced performance. Assigning a unique port to each link prevents this problem and keeps Sense operating reliably.

For example:

- Link 1 can remain on the default **port 9994**.
- Link 2 can be set to **port 9995**.

Be sure to configure the chosen port number on *both* radios at each end of the link so they match. See Fig. 10 & Fig. 11 below on where to set this in the GUI.

For full setup instructions, please refer to the **Advanced Sense Debugging Guide**.

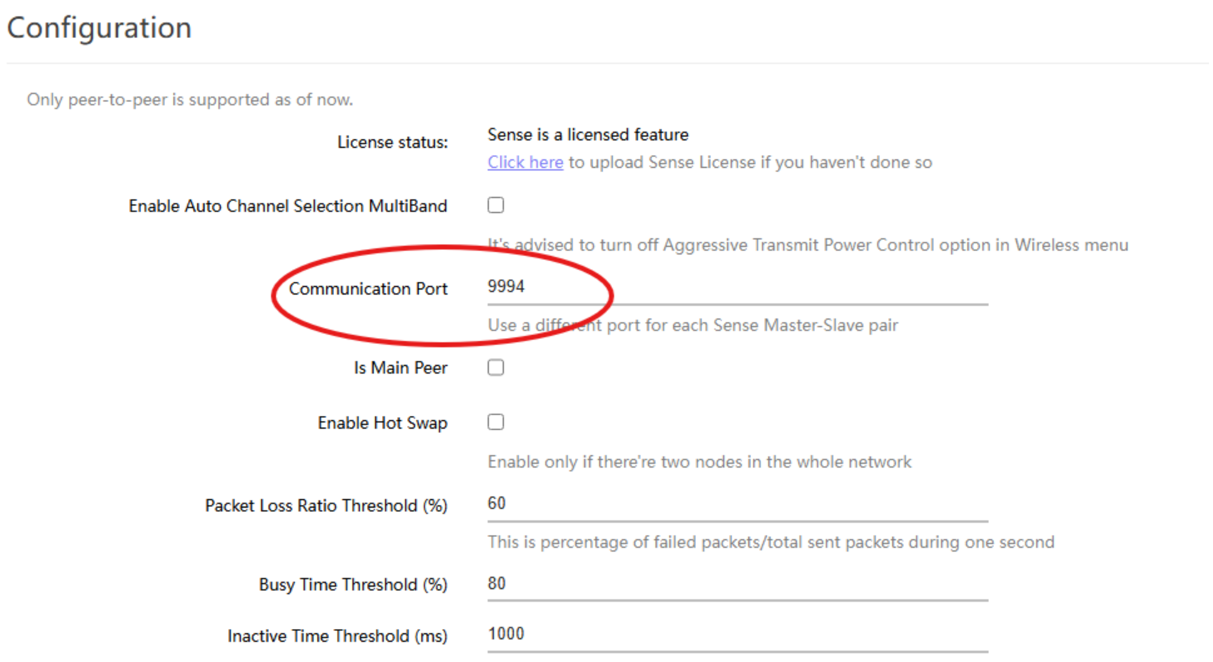


Fig. 10 Communication Port for Sense

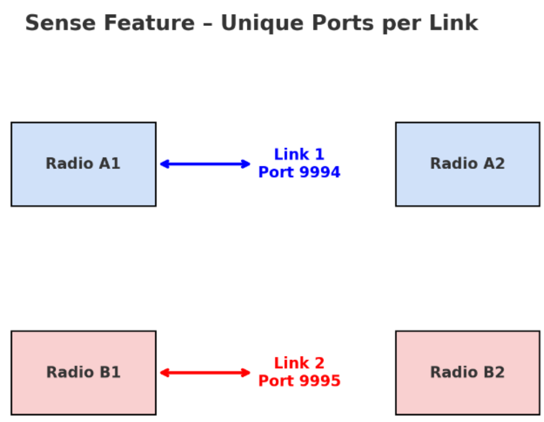


Fig. 11 Sense Port Diagram

Testing & Validation

To ensure and test your Dual Radio system is operating correctly, there are some simple tests that can be run, in addition to observing link metrics that may be useful for troubleshooting.

Ping Check

From any computer or device with a properly configured static IP connected to a radio interface, it should be possible to ping all radios and any other devices connected to interfaces on any other radio.

Ensure you can get a ping response from all radios and devices. To perform a basic ping test simply run ping <Radio IP> from Windows/Linux/MacOS.

Monitor Link Metrics

The most accurate and real-time way to monitor link conditions is by loading **link metrics** directly from each radio.

To do this:

- Open **four** terminal windows — one for each radio in the system.
- Log in to each radio via **SSH**.
- Run the following command on each:

```
ubus listen linkstate
```

This command outputs updates approximately every **4 seconds** (see example output in Fig. 12) in JSON format, including:

- RSSI (signal strength)
- Noise level
- Current frequency in use
- Network activity

```
root@smartradio-301a3bdf6f:~# ubus listen linkstate
{ "linkstate": {"sysinfo":{"cpu_load":[31872,43072,29824],"freemem":78389248,"localtime":1752705155},"oper_chan":55,"ope
r_freq":4595,"chan_width":"0","noise":"-91.293701","activity":0,"lna_status":"3","sta_stats":[{"mac":"00:30:1a:51:07:3e"
,"inactive":90,"rssi":-34,"rssi_ant":[-38,-36],"pl_ratio":0,"tx_bytes":0,"tx_retries":0,"tx_failed":0,"mcs":0}]} }
{ "linkstate": {"sysinfo":{"cpu_load":[29312,42336,29632],"freemem":78393344,"localtime":1752705158},"oper_chan":55,"ope
r_freq":4595,"chan_width":"0","noise":"-91.847763","activity":0,"lna_status":"3","sta_stats":[{"mac":"00:30:1a:51:07:3e"
,"inactive":30,"rssi":-35,"rssi_ant":[-39,-38],"pl_ratio":0,"tx_bytes":0,"tx_retries":0,"tx_failed":0,"mcs":0}]} }
{ "linkstate": {"sysinfo":{"cpu_load":[26944,41632,29472],"freemem":78323712,"localtime":1752705162},"oper_chan":55,"ope
r_freq":4595,"chan_width":"0","noise":"-91.373894","activity":1,"lna_status":"3","sta_stats":[{"mac":"00:30:1a:51:07:3e"
,"inactive":380,"rssi":-33,"rssi_ant":[-37,-35],"pl_ratio":0,"tx_bytes":0,"tx_retries":0,"tx_failed":0,"mcs":0}]} }

root@smartradio-301a3b518d:~# ubus listen linkstate
{ "linkstate": {"sysinfo":{"cpu_load":[31552,45280,30400],"freemem":79503360,"localtime":1754944818},"oper_chan":51,"ope
r_freq":1675,"chan_width":"10","noise":"-92.209648","activity":0,"lna_status":"1","sta_stats":[{"mac":"00:30:1a:3b:51:99
","inactive":60,"rssi":-45,"rssi_ant":[-52,-46],"pl_ratio":0,"tx_bytes":0,"tx_retries":0,"tx_failed":0,"mcs":15}]} }
{ "linkstate": {"sysinfo":{"cpu_load":[29024,44512,30208],"freemem":79544320,"localtime":1754944822},"oper_chan":51,"ope
r_freq":1675,"chan_width":"10","noise":"-93.030518","activity":1,"lna_status":"1","sta_stats":[{"mac":"00:30:1a:3b:51:99
","inactive":110,"rssi":-45,"rssi_ant":[-51,-46],"pl_ratio":0,"tx_bytes":0,"tx_retries":0,"tx_failed":0,"mcs":15}]} }
{ "linkstate": {"sysinfo":{"cpu_load":[37216,45952,30752],"freemem":79769600,"localtime":1754944826},"oper_chan":51,"ope
r_freq":1675,"chan_width":"10","noise":"-91.762413","activity":1,"lna_status":"1","sta_stats":[{"mac":"00:30:1a:3b:51:99
","inactive":50,"rssi":-45,"rssi_ant":[-52,-47],"pl_ratio":0,"tx_bytes":0,"tx_retries":0,"tx_failed":0,"mcs":15}]} }
```

Fig. 12 ubus listen linkstate Running on Two Locally Connected Radios

In a good quality link with good symmetry, each pair of radios will have identical RSSI values on each side, little to no packet loss, minimal noise, and low activity level at idle conditions (not much data passing).

For a complete explanation of each metric, refer to the Link Status Log Fields Explained section in the **Technical Library**.

This method is especially useful for:

- Validating that radios are connected and communicating properly
- Troubleshooting performance or connectivity issues

Note: Doodle Labs will release a real-time graphing utility in Q3 2025 to make this process easier and more visual.

Jamming & Sense Validation

To verify that Sense is working as intended, it's important to create conditions where it needs to respond — this is best done by introducing intentional interference (jamming) to one of the active links.

The goal is to degrade the link quality enough to trigger Sense's automatic channel/band/bandwidth switching and link recovery processes. When done correctly, you should see Sense detect the interference, make a decision, and switch to a better channel from the ACS list.

Basic Steps:

- Follow the interference generation method described in the **Advanced Sense Debug Guide**.
- While the link is being jammed, monitor the Sense debug messages on both radios:
    - `logread -f | grep SENSE`

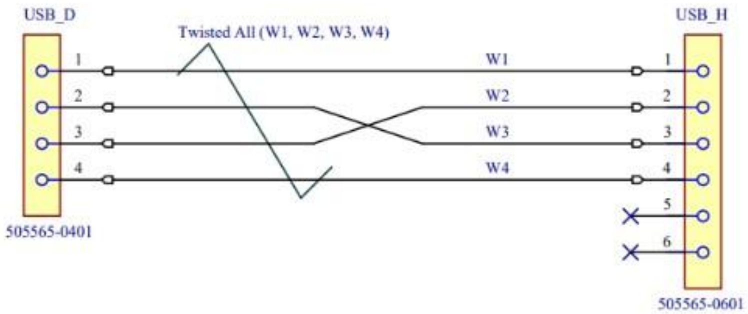
      or
    - `ubus listen linkstate`
- Look for signs of degraded metrics (RSSI, noise, packet loss) and Sense-triggered channel changes.

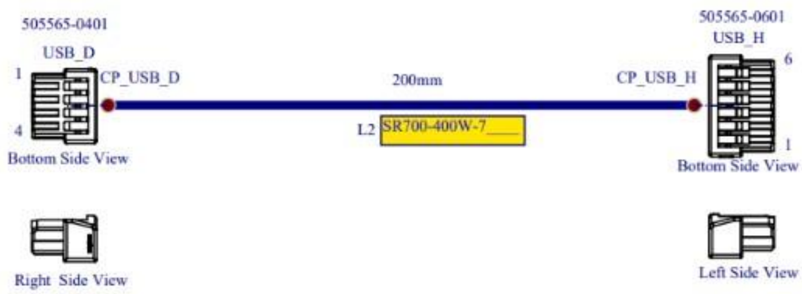
Tip: The debug logs will show when Sense detects poor conditions and initiates a switch, along with the new operating frequency and bandwidth.

Refer to the **Advanced Sense Debug Guide** for detailed interference setup and complete debug command usage.

Appendix

USB to USB cable





Bill Of Materials

Line #	Designator	Name	Quantity
1	L2	LABEL	1
2	USB_D	505565-0401	1
3	USB_D.1.Crimp, USB_D.2.Crimp, USB_D.3.Crimp, USB_D.4.Crimp, USB_H.1.Crimp, USB_H.2.Crimp, USB_H.3.Crimp, USB_H.4.Crimp	505431-1000	8
4	USB_H	505565-0601	1
5	W1	Wires	1
6	W2	Wires	1
7	W3	Wires	1
8	W4	Wires	1