Sense

Introduction

Sense is the name of our upgraded Central Configuration, automatic band/channel/bandwidth selection, and link recovery utility. *Sense* was introduced in the July 2023 Beta Resilience Release and underwent a major update for the September 2023 *Sense* release.

> A license is required to use *Sense's* automatic band/channel/bandwidth selection protocol. Contact a Doodle Labs Sales Representative for more information.

Without a license it is only possible to make manual band/channel/bandwidth changes through either the GUI or the JSON-RPC/UBUS API. This is discussed in further detail below.

Automatic Band/Channel/Bandwidth Selection

The automatic band/channel/bandwidth selection is only enabled with a software license. It consists of two main parts.

1. Link monitoring. The link is constantly monitored, and when the link performance crosses a threshold, the network will switch to a different band/channel/bandwidth.
2. Link recovery. If the link is broken for any reason, the radios will use *Sense's* link recovery protocol to re-establish a connection.

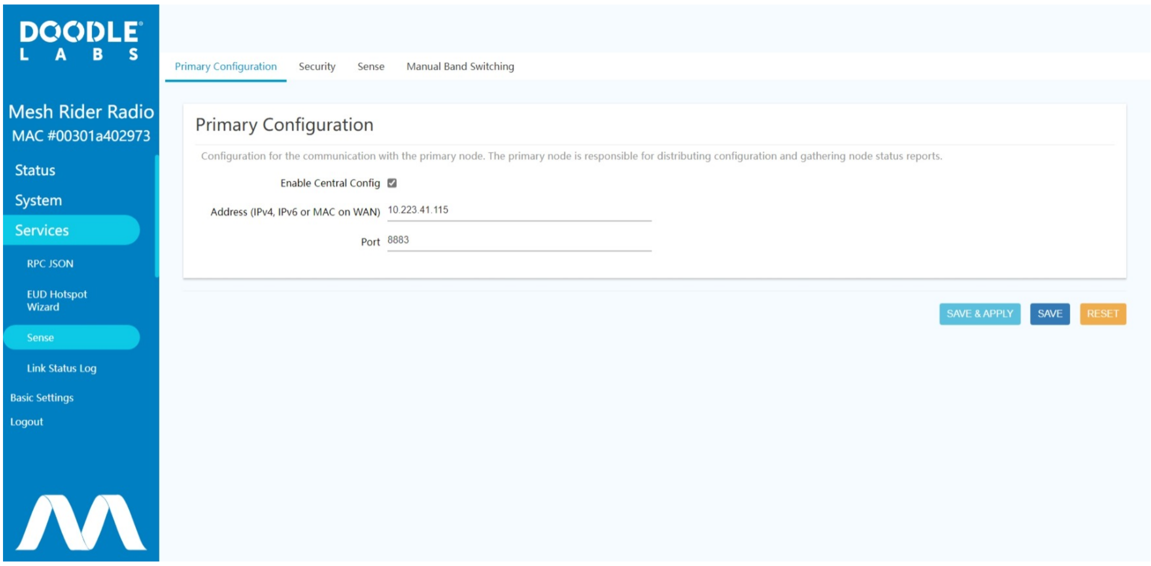
For more information on *Sense* licensing, configuration, and usage, please contact a Doodle Labs Sales Representative.

Manual Band/Channel/Bandwidth Selection

Without a *Sense* license, it is still possible to make manual band/channel/bandwidth changes through either the GUI or the JSON-RPC/UBUS API. If you have a *Sense* license, you can also make manual band/channel/bandwidth changes with *Sense* running.

Setup the messaging system

In the current version, *Sense* uses our Central Config system for messaging, which is based on MQTT. In the messaging system, there is one master or primary node, and there can be multiple clients (currently only one is allowed, or *Sense* will stop). All messages need to go through the primary node, which disseminates the messages to all clients. Navigate to Services -> *Sense* (Fig. 1).



Primary Configuration

Configuration for the communication with the primary node. The primary node is responsible for distributing configuration and gathering node status reports.

Enable Central Config ☑

Address (IPv4, IPv6 or MAC on WAN) 10.223.41.115

Port 8883

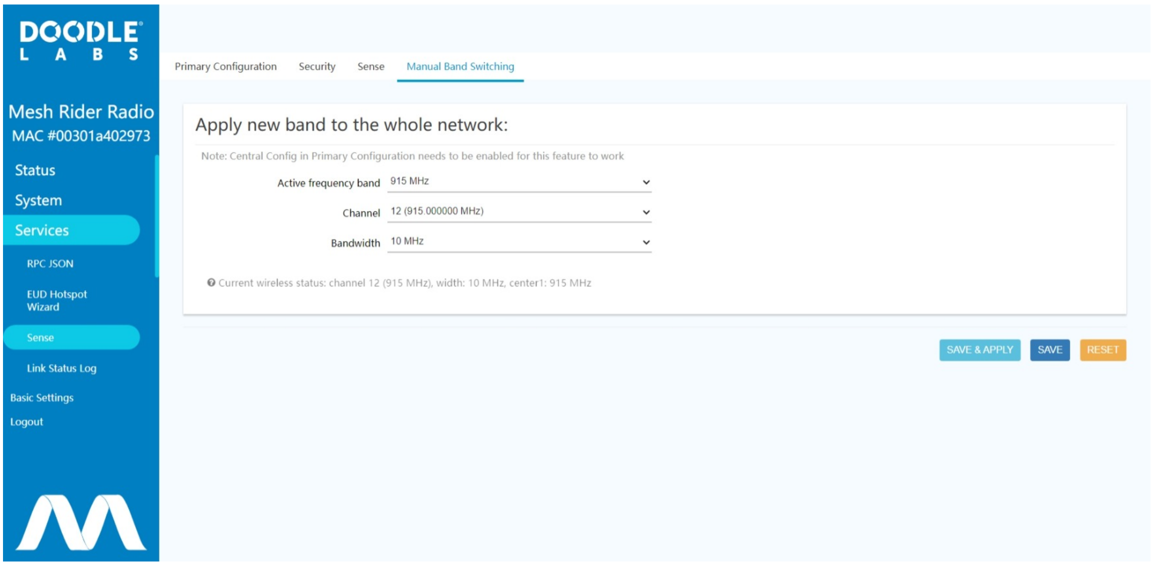
Fig. 1 *Sense* main configuration Menu

The primary node is selected by inputting its address (IPv4, IPv6, or MAC) in the Primary Configuration tab and clicking `Save & Apply`. The *Sense* utility uses TLS for security, and this can be configured in the `Security` tab. The messaging system can also be set up using uci.

```
root@smartradio:/# uci show central_config
central_config.@primary[0]=primary
central_config.@primary[0].enabled='1'
central_config.@primary[0].primary_addr='10.223.41.115'
central_config.@primary[0].primary_port='8883'
```

Changing channels through the GUI

To switch to a different band/channel/bandwidth through the GUI, use the Manual Band Switching tab in the *Sense* menu.



Apply new band to the whole network:

Note: Central Config in Primary Configuration needs to be enabled for this feature to work

Active frequency band: 915 MHz

Channel: 12 (915.000000 MHz)

Bandwidth: 10 MHz

Current wireless status: channel 12 (915 MHz), width: 10 MHz, center1: 915 MHz

Fig. 1 *Sense* Manual Band Switching

Changing channels through the UBUS API

You can also execute a manual band/channel/bandwidth switch using the ubus command below. Ubus commands can be executed remotely over the JSON-RPC API.

```
$ ubus call central-config config '{"dest":"all","delay":0,"config":
{"channel":"50","bandwidth":"5","submodel":"RM-1675-2L-X"}}'
```

Here we are switching to channel 50 with a bandwidth of 5-MHz, using the submodel `RM-1675-2L-X`. You can get a list of valid submodels by running the following command (output shown)

```
$ cat /usr/share/.doodlelabs/fes/$(fes_model.sh get parent)
sub_model0="RM-1675-2L-X"
sub_model1="RM-1815-2L-X"
sub_model2="RM-2065-2L-X"
sub_model3="RM-2245-2L-X"
sub_model4="RM-2350-2L-X"
sub_model5="RM-2455-2L-X"
sub_model6="RM-2450-2L-X"
```

You can check the current band settings using

```
$ iw wlan0 info
Interface wlan0
        ifindex 9
        wdev 0x2
        addr 00:30:1a:4e:bb:02
        type mesh point
        wiphy 0
        channel 51 (1675 MHz), width: 10 MHz, center1: 1675 MHz
        txpower 36.00 dBm
        multicast TXQ:
                qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-
bytes        tx-packets
                0       0       39719   0       0       0       0
3825875         39846
```

In the example above, you can see that the radio is on channel 51 (1675 MHz), with a bandwidth of 10 MHz. You can check the connectivity to other nodes using

```
$ batctl o
[B.A.T.M.A.N. adv 2021.0-openwrt-1, MainIF/MAC: wlan0/00:30:1a:4e:bb:02
(bat0/4a:50:6e:97:51:a7 BATMAN_IV)]
   Originator        last-seen (#/255) Nexthop           [outgoingIF]
 * 00:30:1a:4e:bb:01    0.060s   (246) 00:30:1a:4e:bb:01 [     wlan0]
```

In this case, there is one other radio in the network (00301a4ebb01), and the last packet received from it was 60ms ago. The transmit quality is 246/255 which is very good, and as it is the only other radio in the network, the `Nexthop` is directly to the other radio over the `wlan0` interface.