Lora APRS - VHF APRS Bridge

John Langner, WB2OSZ

First Rough Draft, July 17, 2024

Background

LoRa is a radio technology often used for Internet of Things. It allows long distances with low power at low data speeds. Devices are very inexpensive. A typical board with LoRa, WiFi, Bluetooth, GPS receiver, USB port, battery charger, and display is less than US $50.

Recently there have been many new LoRa APRS projects. They include trackers, digipeaters, and IGates. There is a large chasm between these new projects and the existing VHF APRS network.

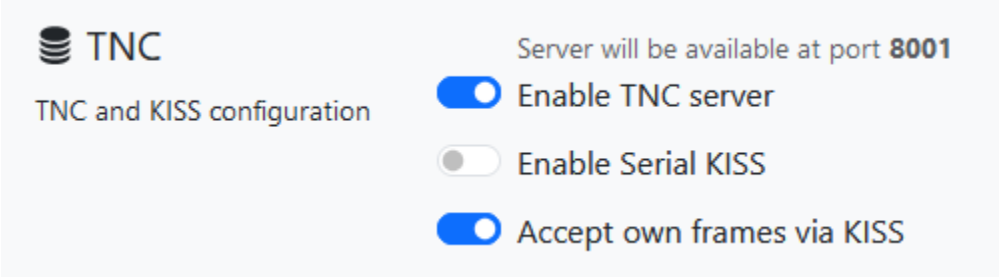
Geoffrey, F4FXL, made a big breakthrough by finding a way to provide bridging between LoRa APRS and traditional APRS. See Building a VHF / LoRa APRS Bridge . Here, is a simpler and more flexible approach.

You Can Play Along at Home

I used the same LoRa APRS implementation but others should also work if they provide a TCP KISS interface.

You will need a digipeater/IGate as described here: iGate/digipeater firmware by CA2RXU.

Attach it to your LAN and be sure to enable these options:



It would be helpful to have a tracker as well so there is something to receive. **LoRa APRS Tracker Firmware** is known to be compatible**.** Be sure that they both use the same frequency and LoRa parameters.

If there is sufficient interest, I will describe how to set up an rsyslog server on a Raspberry Pi. This is useful for troubleshooting.

Test What We Have So Far

In my case, the LoRa Digi/IGate has an address of 192.168.1.238. When playing along at home, substitute your actual IP address.

First, we will run the "kissutil" application that is bundled in with direwolf. Wait until the tracker has a change to transmit a position. You should see something like this:

```
$ kissutil -h 192.168.1.238
[0] WB2OSZ-12>APLRT1,WIDE1-1:!/8wPq<K",[;cQ
[0] WB2OSZ-12>APLRT1,WIDE1-1:!/8wQ#<K"8[<mQ
```

WB 2OSZ-12 is the name of the tracker.

APLRT1 identifies the application generating the packet.

The information part may look a little strange because it using the rarely used compressed format. If you have doubts, you can evaluate it with decode_aprs.

```
$ echo 'WB2OSZ-12>APLRT1,WIDE1-1:!/8wPq<^C,[;cQ' | decode_aprs

WB2OSZ-12>APLRT1,WIDE1-1:!/8wPq<^C,[;cQ
Position, Human, Ricardo, CA2RXU ESP32 LoRa Tracker
N 42 37.1398, W 070 30.3262, alt 39 m (129 ft)
```

See Understanding APRS Packets for more information about demystifying APRS.

Attach it to direwolf

This requires a new feature added for release 1.8. As I write this, 1.8 is not out yet so you will need to use the "dev" branch. Ensure you have "Dire Wolf DEVELOPMENT version 1.8 D" or later. Prepare a direwolf.conf file as shown below. ADEVICE, as shown, is for Windows; change appropriately for your operating system.

```
ADEVICE USB

CHANNEL 0
MYCALL WB2OSZ-14

NCHANNEL 14 192.168.1.238 8001

OBEACON delay=0:15 every=10 via="WIDE1-1,WIDE2-1" sendto=14  symbol="dog"
objname="woofwoof" lat=42^37.17N long=071^20.82W comment="testing - no
comment"
```

Startup direwolf. Notice near the beginning, where channel properties are displayed:

```
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, A+, 44100 sample rate.
Channel 14: Network TNC 192.168.1.238 8001
```

Channel 0 is a normal radio channel using the internal TNC.

Channel 14 is now mapped to a TCP KISS network TNC. It can be treated like the normal channels for beaconing, digipeating, IGate, and packet filtering.

When we beacon to channel 14 (sendto=14), it is sent to the network TNC, i.e. the LoRa device.

```
[14>nt] WB2OSZ-14>APDW18,WIDE1-1,WIDE2-1:;woofwoof
*010000z4237.17N/07120.82Wptesting - no comment
```

If you have rsyslog running, you should see:

```
… …  WB2OSZ-11 CA2RXU_LoRa_iGate_1.3 TX / WB2OSZ-14>APDW18,WIDE1-1,WIDE2-
1:;woofwoof *010000z4237.17N/07120.82Wptesting - no comment
```

WB2OSZ-11 is the name of the LoRa Digi/IGate. "TX" indicates that it is transmitting.

It works in the other direction as well. We see the tracker (WB2OSZ-12) reception in the syslog.

```
… … WB2OSZ-11 CA2RXU_LoRa_iGate_1.3 RX / GPS / WB2OSZ-12 / APLRT1 / WIDE1-1 /
-87dBm / 11.50dB / -133Hz / 42.61898N / -71.34720E / 0.0km
```

Direwolf displays:

```
[14] WB2OSZ-12>APLRT1,WIDE1-1:!/8wQ?<K"$[<4Q
Position, Human, Ricardo, CA2RXU ESP32 LoRa Tracker
N 42 37.1334, W 071 20.8400, alt 43 m (141 ft)
```

Build a Bridge

Next, we want to forward packets from LoRa to VHF and vice versa. Add this to the configuration file.

```
DIGIPEAT 14 0 ^WIDE[3-7]-[1-7]$|^TEST$ ^WIDE[12]-[12]$

DIGIPEAT 0 14 ^WIDE[3-7]-[1-7]$|^TEST$ ^WIDE[12]-[12]$
```

Example: Received on LoRa, forwarded to VHF APRS.

```
[14] WB2OSZ-12>APLRT1,WIDE1-1:!/8wQE<K"=[=gQ
Position, Human, Ricardo, CA2RXU ESP32 LoRa Tracker
N 42 37.1324, W 071 20.8321, alt 57 m (187 ft)

[0L] WB2OSZ-12>APLRT1,WB2OSZ-14*:!/8wQE<K"=[=gQ
```

[14] means it was received over LoRa and mapped to channel 14.

[0L] means it was cross-band digipeated onto VHF.

You can also perform the LoRa APRS digipeating to take advantage of the greater functionality provided by direwolf.

```
DIGIPEAT 14 14 ^WIDE[3-7]-[1-7]$|^TEST$ ^WIDE[12]-[12]$
```

Very flexible filtering is available to place any desired restrictions on what gets forwarded.

```
FILTER  14  0  … …
```

See APRS Digipeaters for more information about digipeaters.

Let direwolf Perform the IGate Function for LoRa

To be continued.

See Successful APRS IGate Operation for more information about APRS IGates.

Add Your Favorite APRS Display Application

Configure your favorite APRS application (APRSIS-32, YAAC, PinPoint APRS, Xastir, etc.) to use direwolf as the network TNC. Packets from both the VHF network(s) and the LoRa network(s) will all show up on the same map.

Other Uses

This new network TNC feature is not limited to LoRa. It could be used with other modem types such as fldigi, M17, VARA, etc., and others not invented yet, as long as a network TCP KISS interface is available.