43 - Mesh network

Jump to bottom

Nunu edited this page on Mar 2 · 12 revisions

NUNU Protocol

This firmware since `v.21.0` incorporates message hopping mesh network functionality called the **"NUNU Protocol"**.

> Tip
>
> You can download early beta NUNU Protocol release here.

Which allows to extend the range of infrastructure-less communications via intermediate stations (nodes).

This way of communications can also be described as flooding.

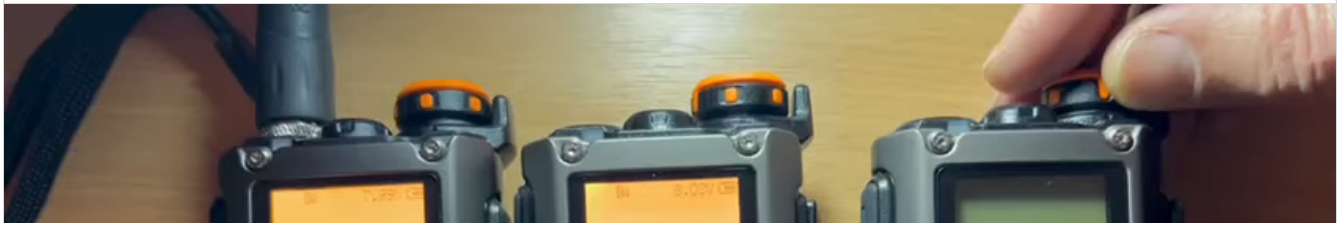
The protocol is based on the Meshtastic principles.

Videos

For a quick explanation of Meshtastic please check out video below:

Video of the NUNU Protocol showing message hopping with 3 radios:

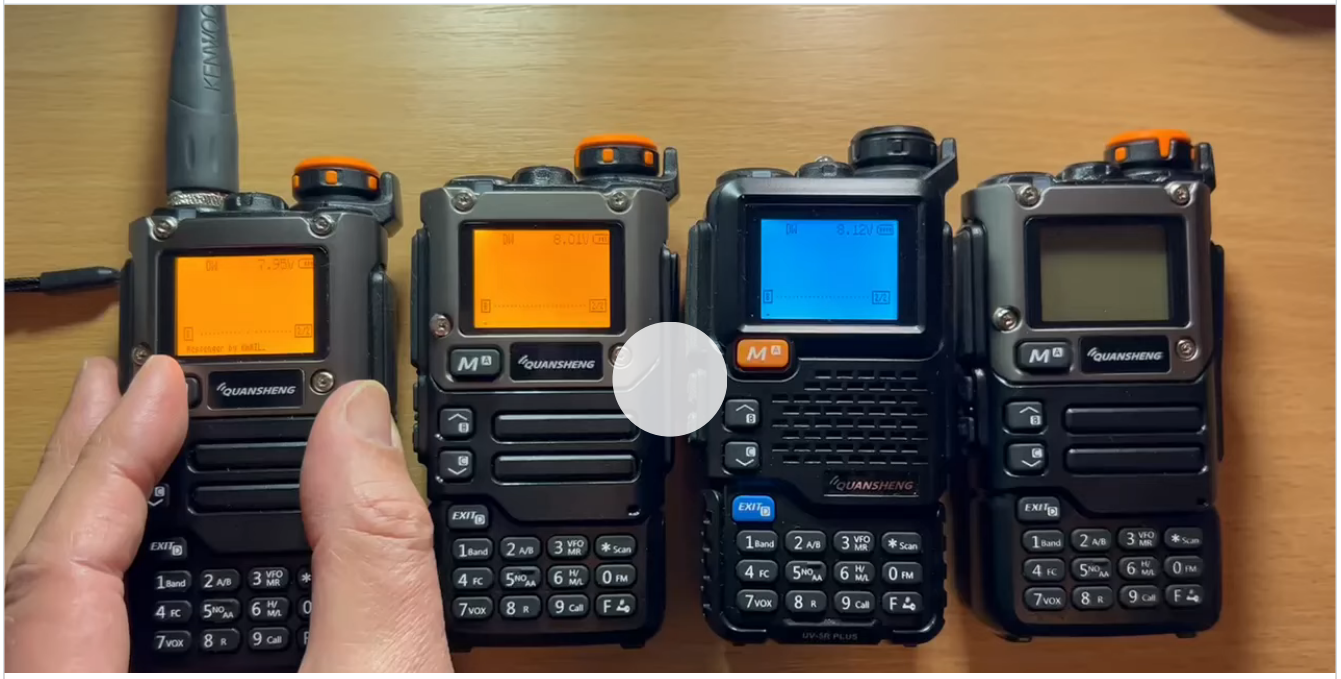
hop.mp4



0:00 / 0:15

Video of the NUNU Protocol with 4 radios:

hop2.mp4



0:00 / 0:16

Functionalities

NUNU Protocol implements:

- Carrier-Sense Multiple Access with Collision Avoidance
  - the radio will not TX if the line is busy to avoid collisions
  - if the line is busy the radio will wait with TX for the random amount of time, until the line is free
  - of course there might be a case when 2 radios TX at the same time because they didn't hear each other (also known as the hidden node problem).
- CRC codes to determine if packet has been corrupted in transit

- CRC codes to determine if packet has been corrupted in transit
  - all messages with incorrect CRC codes are ignored
  - this is a vital part of the protocol as it prevents hopping incorrect packets through the mesh network
- Give and take approach
  - Users can only request that their messages be hopped if they contribute to the network by hopping messages for others
  - controlled via the `MsgAck` menu option
- Encryption is supported
  - The nodes that receive encrypted packets without having the right encryption key will not understand them, but will still hop them forward
  - This enables communities to share the mesh network and still have ability to have both public and private conversations
- Hopping state which once entered
  - drops all incoming packets
  - prevents user sending new messages

Packet structure

Packet structure:

Size (in bytes)	Name	Description
4	sync word	Sync word for this FW `0x30 0x72 0x57 0x6C`
1	header	Packet header
30	payload	Packet payload (can be encrypted)
13	nonce	Packet nonce used for encryption
8	crc	Error detecting code (based on the display hash function)

Header structure:

Size (in bits)	Name	Description
5	type	Header type
3	hop	Hop count (max 7 hops)

Header types:

Value	Name	Description

0	MESSAGE_PACKET	Unencrypted message
1	ENCRYPTED_MESSAGE_PACKET	Encrypted message
2	INVALID_PACKET	Invalid packet, headers greater and equal to this one as treated as invalid packets

Clone this wiki locally

```
https://github.com/kamilsss655/uv-k5-firmware-custom.wiki.git
```