How does M17 encode my voice?

M17 is a digital mode

- That means we are handling ones and zeroes
- Speech is analog

Sampling the microphone signal

- That’s the first step of ‘digitizing’ voice

Sampling the microphone signal

- Speech is sampled at 8 kHz, it’s a standard rate
- For best results, the A/D converter’s resolution should be 16 bits
- One speech subframe lasts 20 milliseconds
- 2 subframes are concatenated to form a 40ms frame

Sampling the microphone signal

- Subframes don’t overlap each other
- Speech samples are encoded using Codec2
- Depending on the mode, either 1600 (half rate) or 3200 bits per second (full rate) is used

Sampling the microphone signal

- The result is an array of either 64 or 128 bits, depending on the codec rate

Hello, who’s there?

- To send voice and/or data over RF (or any other medium), the M17 encoder has to know a few things
- That information is: destination, source, type of data and metadata

The Source and Destination

- Both fields are 48 bit long
- They usually contain encoded callsigns
- Some values are reserved
- String encoding scheme is simple
- M17 doesn’t rely on any external user ID database

SRC and DST encoding

- We treat the callsign (or any ASCII string) as base-40 number
- Spaces are prohibited

SRC and DST encoding

Character	Value
A .. Z	1 .. 26
0 .. 9	27 .. 36
- (dash)	37
/ (slash)	38
. (dot)	39

SRC and DST encoding

- Encoding works right-to-left (just like the decimal number system!)
- Spaces and illegal characters that are not in the alphabet are ommitted

Message content

- The TYPE field is 16 bits long
- It tells users what type of transmission is about to begin (or is actually ongoing – more details later)

Message content

- Is it a packet data or a stream?
- Is it voice, data or both, interleaved?
- Is it encrypted? If yes – how?
- What’s the Channel Access Number (CAN)?

Message content - CAN

- Channel Access Number is pretty much an equivalent of Color Code known in DMR
- ...or CTCSS/DCS in analog systems
- There are 16 different CANs to pick from (0..15)

The Metadata

- 112 bits long field
- Suitable for cryptographic metadata like IVs or non-crypto metadata like the sender's GNSS position
- Can be used for low-speed data transfer along full-rate voice

Can we start the transmission?

- We know how to encode voice using Codec2
- We know who’s calling who and how to signal that to other users
- We also know the transmission contents

Late joiners!

- What if someone started listening to the transmission half-way?
- We need a way to spread out the link information data (destination, source, type and metadata) over the whole transmission

Link Information Channel

- ...or LICH for short
- It’s a dedicated part of a frame that holds a chunk of the whole link information data set
- We split the initial data set into 6 chunks and transmit them in a round-robin fashion in every frame

Link Information Channel

- That way we ensure every user can reconstruct the original data set after receiving 6 frames
- That’s 6*40=240 milliseconds
- We are not leaving late-joiners behind!

What about the „early” joiners?

- The original, full link information data set is sent in a special frame that preceeds any data transfer over the RF link
- We called that frame the Link Setup Frame
- It’s also 40 milliseconds long

Actual voice frames

- We know how to tell users about the content of our message
- But how do we send Codec2 encoded speech?

Actual voice frames

- Frames are 40 milliseconds long
- They contain one sixth of the LSF (48 bits)
- ...the Frame Number (16 bits)
- ...128 bits of payload (Codec2 data goes here!)
- ...4 bits to flush the convolutional encoder (we’ll get back to this later)

Actual voice frames - sync

- Every frame is preceded with a synchronization vector (pattern) that is 16 bits long
- The purpose of it is to ensure frame synchronization at the receiver
- There are 4 different syncwords, depending on what the link content or type is

Actual voice frames - sync

- Sync vectors are designed to have lowest cross-correlation possible to avoid mistakes

M17 starts with a beep

- Before any transmission is started, we need a way to ‘warm up’ the RF (physical) part of the link
- A preamble is used for this exact purpose
- You’ve guessed that right – it’s also 40 milliseconds long*

M17 starts with a beep

- The preamble is just a 2.4 kHz tone*
- The LSF and voice frames are transmitted immediately afterwards (LSF is transmitted once)
- Preamble helps in receiver synchronization
- Can be used to make sure the PA ramped up to its nominal power

Speech to bytes - summary

- After the PTT is pressed, the preamble is sent out
- The LSF comes next (once) with its own syncword
- A stream of frames continues

Stream end marker (EOS)

- End Of Stream is signalled using a repeating pattern of symbols
- All receiving parties can cease decoding when that pattern is received

73!