Nova Firmwave Outline
Paul Richards no@sosgez.co.uk

Overall Structure
See Fig 1 sketch.

Input
N(=9) off digitised 2 channel sound samplers, probably on 2 diaphragm microphones. Data encrypted.

Output
1 compressed multiplexed stream for IP on 4G transmission to server. Data encrypted.

Assumptions
- You mentioned "12s". I take that to mean a 12 second measuremt window, or epoch.
- Epochs could be isolated or back-to back.
- It appears you chose not to put hardware codecs at the samplers.
- Each sampling hardware unit is recording into its local memory, continuously. Either it has sample rate and gain controls from the controller, or automatic gain.
- The systems regularly achieve a time sync across all samplers by BLE broadcast reference to avoid drift.
- I assume you don't want a hardware codec at the sensors.
- So we will probably use a stream encoding compression system like an **AAC** algorithm, concatenate or interleave at least 9 channels. A 10:1 compression could be be lossy, and we might create a proprietry streaming protocol to package the multiple channels the data is sent in, in structured blocks on TCP/IP adding extra non-audio data. It would be separated and reconstituted at the server then re-

analysed for feature extraction, reporting, logging.

- I shall use "1D" to represent a scheme with small time period coding part of a wave. "2D" coding considers the whole epoch and an expensive transformation such as FFT. Sketch of both in Fig 2.
- Incoming data is already encrypted at the microphone sampler end. Eavesdroppers will know there is a lot of traffic but not read it.
- Outgong data is encrypted
- The server is multi-cored and doesn't need to receive one channel data set before another.
- Server connect via LTE is probably a nearby phone or module with BLE control.

Key Processes

Seq means a sequential list of operations.
Par is a paralell list of operations, loaded across available tasks and cores.

```
Seq
  Connect to all sensors by BLE.
  Connect User Equipment to LTE.
  Request Service level and resources and authenticate
  Loop epochs:
      Sync       Per epoch, synchronisation from controller to
                 all samplers by broadcast.
      Par:
           Receive
                 Par N channels tasks:
                       Seq:
                             BLE Request data in smallish
                                   chunks by giving time window
                             relative to sync
                             Request filler blocks as needed
```

Decrypt
Optionally filter both channels.
Scale and phase shift Background relative to Foreground channel.
Noise Reduction - Form F-B
Compress using AAC in 1D.
(2D e.g. DWT FFT could be beyond the hardware budget!)
Encrypt data and coefficients is used.
Chop into conventiant transport chunks and add framing
Enqueue data
Enqueue statistics/debug/status

Send
Dequeue
Fine chop (LTE kit probably does this)
Physical send 4G hardware, again on BLE.
Respond to resend requests to infill

SelfTest/Stats
Periodic self test and performance stats
Encode
Enqueue

Description

Per-channel reception process request chunks on an available BLE device from the samplers. Idealy the controller acquires all data by 1 chunk later than end of epoch. i.e, chan1chunk1, chan2chunk1,.. chan9chunk1,chan1chunk2...
These fill up a receive buffer frame but we might lose some chunks due to poor comms so there much be a mechanism to request resending of data, with the possibility of giving up

and substituting fallback bland data if thats better than no data.

We must be able to gather data faster than it was recorded (44kHz), to allow some channel capacity to request infills.

There could be a priority across the channels. e.g. channel 5 gives more significant data than channel 8.

We will be receiving with a small latency during the sampling period, and NOT waiting until the full sample is complete. Each chunk can be decrypted as it is independant.

Raw samples at 44K might be hissy anyway, so a bit of low pass software filtering could be useful, before subtracting background from foreground. That assumes we have the right relative amplitude and phase shift, if the foreground and background mics are far apart. Out[t[= F[t]-k*B[t+p]

I don't know why you prefer to do this at the control side. Maybe you would let the samplers do it instead?

If we use a 2D compression we will need a full epoch of data in a channel to start a calculation. This is likely to be lossy for such a high compression ratio,10:1. Whichever method, it might take several tests to identify which coding is best suited since the data we gather is not common like voice signal. The variable sized amounts of output data could mean the channels complete tasks in random order.

Picking the right hardware is important as it speeds up intensivly computational sections a lot. e.g. many cores with FPU, hardware encryption instructions, e.g. AES.

Composite output data rate is something like 1MByte per sec down from @10MB on input.

In the background the system would need features for monitoring its own quality, battery status, controlling parameters. In future developments you might want to add other sensors/MEMS devices too.

e.g. sending data in the clear, parameter controls,selecting auto gain controls, detecting when capacity for computation or comms is saturated. So the output stream is going to be a mixture of data objects, not just sounds.

Hardware/Software Suggestions

More than 1 BLE physical connection could be employed simultaneously to reduce latency or cope with unreliable links from samplers. The hardware cost is trivial compared to operational costs with all that data flying about.. .

If the idea is get the server to reconstrct the cleaned 9 sound channels, and then process it to extract features, start with no compression/codec and find out what quality of data you really need to be able to make good diagnoses. You might discover that fine detail MUST be preserved, so you must employ a lossless codec.

It might also reveal that you don't need to send all this data to the server because you can undertake the analysis locally and upload much shorter reports. Give the ability to flip between working scenarios. Highlighted events can cause raw data retention for additional evaluation by the consulatnt.

PR