OsmoDevCon 2012

Sylvain Munaut

28C3, March 23rd, 2012

Osmo GMR

Topics

- Voice codec
- TDMA frame work
- MultiRX tool

Voice codec
The problem

- Proprietary AMBE variant of DVSI
- Not supported by their "cheap" USB sticks
- Cheapest hw is their NET-2000 appliance. (and the 2000 part is the price in EUR :)
- No public specs

Voice codec
What we have

- 4 frame types and how to differentiate them
  - Tone
  - Erasure (i.e. repeat last)
  - Silence
  - Voice
- Tone frame format is known
  - Mostly ... the amplitude scale is weird
- Voice frame bit allocation is mentionned
  - Class 1: 9 pitchs bits, 6 gain bits, 6 voicing bits, 27 spectral bits
  - Class 2: 2 gain bits, 30 spectral bits
- We got P25 IMBE and AMBE specs from their "public" ftp.
- We have the patents

Voice codec
What we don't have

- The rest ...

TDMA framework: General thoughts

- TODO

MultiRX tool
Introduction

- Captures multiple channels simultaneously
- Writes channelized data to a file (or fifo)
- Takes care of all the multirate parameters calculation
- Easy to use: ./gmr_multi_rx –gmr-dl ARFCN ARFCN ...
- Currently supports libusrp, UHD, FCD

MultiRX tool

Ongoing development

- Abstraction library for various sources
  - FCD, UHD, OsmoSDR, RTL-SDR, ...
- Support for more channel configurations
  - DECT, TETRA, GSM, APCO P25, MPT TODO,
- More sophisticated chanelizer design
  - Distribute requested channels over available sources
  - Use of polyphase filterbank if possible (e.g. !TETRA)
- File import for various IQ formats
  - Relative channel mode using +/- prefix

MultiRX tool
Outlook

- Discovery (Scanning) mode
  - On the fly reconfiguration of the flowgraph
- Integration of available
  - Allows better signal acquisition
- GRGPU?
- More hardware QA
- GUI

GSMTap v3

Motivation
Why change at all ?

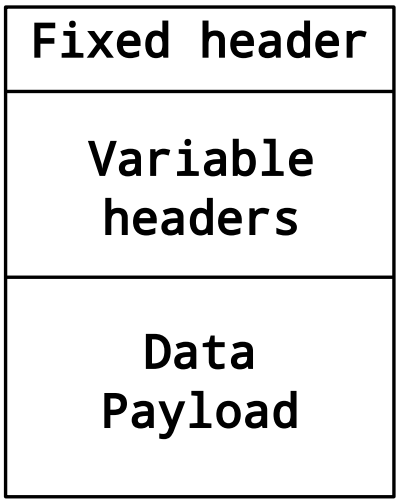
- GSMTap is great, it has served us well over the years.
- But as we extend (abuse) it, its limitations becomes more and more obvious
  - After all it's been designed for GSM
- We need a new revision of GSMTap.
- The idea has been raised before.
- I think it's time we make that happen now !

Requirements

- Extensible
  - We won't think of everything the first time around
  - Need to easily add new Protocols, Channels types, Info Fields, ...
- Specified
  - You know ... a bit more than a single `.h` file
  - And hopefully allow distribution of *authority*
  - Anybody wants to write an `RFC` ?
- Easy to use API
  - Reference implementation in `libosmocore`
  - But concepts should translate to other frameworks / languages
- Compatibility path for old app ?

Specifications: General Idea

- Divided in 3 zones:
  - Fixed Header
  - Variable Headers
  - Data Payload
- All headers aligned on 32 bits word boundaries
- All fields in network byte order
- Must fit in a single UDP packet
  - If you must, you could always use IP fragments ...



Specifications: Fixed header

- Common part
- Mostly compatible with `GSMTapv2`
- Dictates how the "variable" part is interpreted
- Format: 1 single 32 bit word = 4*8 bits fields
  - `version`: fixed to '3'
  - `hdr_len`: Total header len in 32 bits words unit
  - `type`: Main type
  - `subtype`: Subtype if applicable, fixed to 0 if not.
- The 'type' field could be used as an *authority* split point

Specifications: Variable headers

- Will mostly be dependent of the protocol
- But a few could be protocol agnostic (comments / ...)
- Some can be mandatory / optional
- Use TLV so that unknown headers can be skipped
- Format:
  - `tag`: (8bits)
  - `len`: (8bits) In 32 bits words unit
  - Rest is payload
- Tag address space split:
  - `0x00 -> 0x3f`: Protocol agnostic
  - `0x40 -> 0x7f`: Reserved
  - `0x80 -> 0xff`: Protocol dependent
- Inside a block we could tolerate optional values (using and invalid marker). This would avoid too many different blocks.

Specifications: Data payload

- Whatever you want to encapsulate in the first place
- No restrictions except it must fit in a single packet

Specifications
Example

A few quick examples of variable headers out of the top of my mind about GSM

- **Stream Ident**: Unique ID provided by the app to differentiate streams easily
- **Logical channel info**: chan_nr / subslots / tn / ...
- **Physical channel info**: ARFCN / ...
- **Timing info**: Frame numer + Epoch
- **Data type**: For raw burts. Hard bit / Soft bits / ...

API

- `void *osmo_gsmtap3_add_hdr(struct msgb *msg, uint8_t tag, unsigned int len)`
- `void *osmo_gsmtap3_add_data(struct msgb *msg, unsigned int data_len)`
  - Required to be called last !
- `void *osmo_gsmtap3_get_hdr(struct msgb *msg, uint8_t tag, unsigned int len)`
  - Require the length as a check against malformed packets
- `void *osmo_gsmtap3_get_data(struct msgb *msg, unsigned int *data_len)`

Open Questions

- How to deal with info that changes across a single payload
  - Example would be `ARFCN` for L2 frames on a hopping channel
  - Repeat the header block: Makes API more complex ...
  - Have a different "L2 Physical Info" block with 4 values ?
  - Ignore it and just spec to take the first/any value ?
- Should the data zone even be separate or just another header/TLV block ?
- Transition plan ?
  - Do we need one, or just ... do it as quick as possible and let people upgrade.
- ...