

OsmoMGW - Feature #4535

Develop "E1/Abis generator"

05/05/2020 07:02 PM - laforge

Status:	In Progress	Start date:	05/05/2020
Priority:	High	Due date:	
Assignee:	laforge	% Done:	60%
Category:			
Target version:			
Spec Reference:			
Description			
This will then provide a E1 line with 16k (and possibly 8k) sub-slots and TRAU frames for the different voice codecs via libosmo-abis. It will enable development and testing of osmo-mgw without any external dependencies. I am also considering to include a RTP to E1 conversion inside that generator, so that a test caes could send an RTP packet to the e1-gen, which would turn it into a TRAU frame that appears on a sub-slot of an E1 timeslot, which then subsequently travels into osmo-mgw, which will perform the inverse operation and re-create a RTP packet. The test suite could then compare the two RTP packets. Details will be worked out on-the-fly. Right now I'm mainly workin at the I.460 subchannel mux/deumux as well as TRAU frame alignment for all of the modes/codecs.			
Related issues:			
Precedes OsmoMGW - Feature #2547: Add E1/T1 endpoint type to osmo-mgw		Resolved	05/06/2020 05/06/2020

History

#1 - 05/05/2020 07:03 PM - laforge

- Precedes Feature #2547: Add E1/T1 endpoint type to osmo-mgw added

#2 - 06/13/2020 06:23 PM - laforge

- % Done changed from 0 to 20

some updates.

- I.460 multiplex doing 16k but also 8k sub-slots implemented in <https://gerrit.osmocom.org/c/libosmocore/+/18247> (merged)
- new API for TRAU frame parsing/encoding: <https://gerrit.osmocom.org/c/libosmo-abis/+/18249> (WIP)
- new code for TRAU frame synchronization: <https://gerrit.osmocom.org/c/libosmo-abis/+/18250/5> (WIP)
- new code for TRAU frame to RTP conveersion: <https://gerrit.osmocom.org/c/libosmo-abis/+/18382/3> (WIP)

#3 - 06/13/2020 08:27 PM - laforge

- Checklist item [] test whole chain using real-world FR + EFR traces added

Build verification of all patches passes now. Next would be some kind of verification.

I'm planning to fire up my good old BS-11 tomorrow, set up a call and do a capture of at least FR and EFR on 16k TRAU sub-slots. This test data can then be used for an end-to-end test through the full chain of I.460, trau_sync, trau_parse, trau2rtp and the resulting RTP played back for verification.

#4 - 06/21/2020 03:27 PM - laforge

- % Done changed from 20 to 60

I've just pushed an update to the laforge/trau branch of libosmo-abis, which contains the new code modules for E1 / TRAU usage, and also a small test program and test data.

The test program takes a raw binary dump created from an E1 timeslot with TRAU frames captured of a BS-11 and converts that to RTP packets. You can feed those RTP packets into osmo-gapk to play them back. The audio is clear, there are just a few seconds of quiet at the beginning before anything is audible.

The patches also add an "I460" mode to e1_input. So from the osmo-mgw side, I think we need to

- link against libosmo-abis and call `e1inp_init + e1inp_vty_init()`
- use existing VTY code to configure E1 line numbers / drivers / ports
- call `e1inp_ts_config_i460()` on every timeslot we want to use (this should **not** happen on all timeslots, as some of them are used for signalling by the BSC)
- call `osmo_i460_subchan_add()` / `osmo_i460_subchan_del()` for each i.460 sub-slot we want to use
- build the processing chain like in `libosmo-abis/contrib/trau2rtp/trau2rtp.c` via `i460 -> trau_sync -> osmo_trau_frame_decode_16k/osmo_trau_frame_decode_8k -> osmo_trau2rtp`

I believe this should be at a stage where it can be integrated into osmo-mgw. The API may not be 100% stable yet, but it shouldn't change completely. The main missing bits are:

- allow selection of sync pattern in `trau_sync.c` (or auto-detect?)
- handling of T-bits for alignment of air-interface timing with downlink TRAU frame timing

I will next be hacking on a small tool that will speak the 'osmo-e1d' protocol but use binary capture files that are looped over and over again. This way you should be able to use libosmo-abis `e1_input` without any physical card and still get TRAU frame data.

At a lower priority, I'd like to look into support for HRv1 and AMR TRAU frames on 16k sub-slots.

#5 - 06/30/2020 05:35 PM - laforge

my most recent work has been in [osmo-e1d](#), adding support for a "virtual E1 interface pair".

In `e1d` language, an "interface" can have multiple lines (e.g. a 4-port interface would have 4 E1 lines, each having 32 timeslots).

The so-called "vpair" driver for `osmo-e1d` will create two virtual E1 interfaces with each an identical number of lines. The lines will be mapped 1:1 and provide something like a "virtual E1 loopback cable". So you can have two applications written for talking to E1 lines (e.g. via libosmo-abis) but without any hardware and a physical cable in between.

I've also written a `osmo-e1d-pipe` command line program which can be used to connect `stdin/stdout` of the program with an E1 timeslot. It has the capability to read a file containing a raw timeslot dump and transmit that via a E1 timeslot. It will loop after EOF.

So putting things together, we now can build a system where a capture (such as my 16k TRAU captures for FR and EFR taken on a BS-11) can be played back into one side of a vpair, and the other side can be opened by software (the future osmo-mgw) and it will receive an endless loop of recorded TRAU frames for testing.

Likewise, we can use this as a vehicle in our TTCN3 tests, where the TTCN3 test would need to get a test port that talks to `osmo-e1d` in order to send TRAU frames or other E1 streams to the IUT (`osmo-mgw`, `osmo-bsc`).

I'll put some documentation to the [osmo-e1d](#) project wiki.