Core testing infrastructure - Feature #4692

finally implement testing of the actual voice audio path

08/05/2020 08:27 AM - laforge

Status:	In Progress	**Start date:**	08/05/2020
Priority:	Normal	**Due date:**	
Assignee:	laforge	**% Done:**	40%
Category:			
Target version:			
Spec Reference:			

Description

For many years we've set up tons of testing, but it's all on the control plane, and the user plane for PS services. We don't yet have any testing of the acutal voice path, so we wouldn't detect codec problems drop-outs or whatever other voice quality issues in any of our testing.

History

#1 - 08/05/2020 09:12 AM - laforge

- File simcom-tone.c added

- Status changed from New to Stalled

- % Done changed from 0 to 10

There are very few modems that support exposing the audio data in any other format than an electrical PCM interface, see my blog post form 2017 at https://laforge.gnumonks.org/blog/20170902-cellular_modems-voice/

The situation unfortunately didn't change meanwhile, so we still are stuck with that.

The only vendor that persistently supports PCM audio over USB (at least according to technical documentation) is SIMCOM. I've been trying to get it to work, but unfortuantely failed so far.

I can establish the voice call and issue the AT+CPCMREG=1. From that moment on, there also is data coming out of the /dev/ttyUSB device, but the data rate is much too low, and it's very bursty. Interestingly, the amount of data readable from that /dev/ttyUSB device depends on whether or not there is voice activity in the channel, i.e. during quiet phases, there is very little data, and when there is voice data from the remote end, there is more data to read. This is not at all what one would expect from a S16LE 8kHz PCM sample stream.

When looping the data (e.g. with dd if=/dev/ttyUSB9 of=/dev/ttyUSB9), all I get in the audio channel is some strange noise.

Likewise, http://settrans.net/~rah/misc/simcom-tone.c doesn't play any tone to the remote side, nor does it record any audio.

I currently only have a SIM7230E available, which is already discontinued. The fact that SIMcom doesn't seem to be making any firmware updates available doesn't help the situation, either.

I guess I'll wait for some other SIMcom modem models to arrive, which I ordered two days ago.

#2 - 08/05/2020 09:14 AM - laforge

There also still is the QMOD-BGS2 approach that we started at sysmocom some time ago, which is basically using a XMOS to implement multiple PCM slave interfaces. It requires significant software R&D and I'm looking for a somewhat faster/simpler approach first. If there's no success with SIMcom, we can fall back to that.

#3 - 08/06/2020 08:42 PM - laforge

- Status changed from Stalled to In Progress

- % Done changed from 10 to 40

Ok, good news is that with a SIM5360E I could actually get things to work. It's a bit more intricate than with the SIM7xxx (even per documentation), as there are the following two steps required:

- Switch one of the usb-serial devices from DIAG mode to DATA mode by means of AT+DSWITCH=1
- Actually enable the PCM on the DATA ttyUSB using a binary sequence of bytes

Details see atthached document.

As for the actual audio payload, I've implemented a RTP bridge in https://git.osmocom.org/osmo-mgw/log/?h=laforge/simcom

Now the question is how to proceed further, given that we have RTP (PCMA) on the MS side and RTP (gsm specific codecs) on the CN side.

Files

simcom-tone.c	5.73 KB	08/05/2020	laforge