

OsmoBTS - Bug #2700

Odd RTP behavior in case of bad / missing speech frames

12/02/2017 12:54 PM - laforge

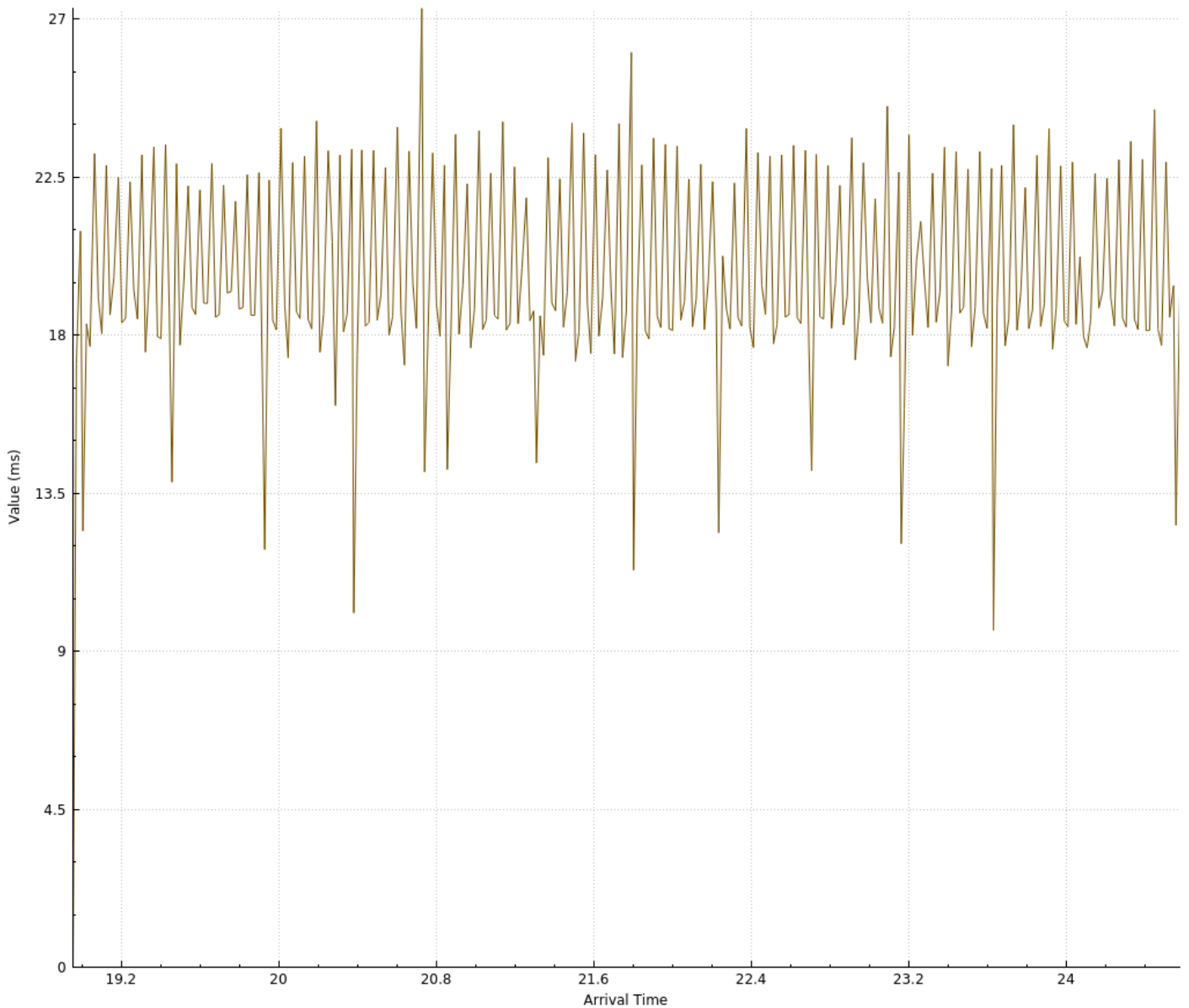
Status:	Closed	Start date:	12/02/2017
Priority:	High	Due date:	
Assignee:	pespin	% Done:	70%
Category:	osmo-bts-trx		
Target version:			
Spec Reference:			

Description

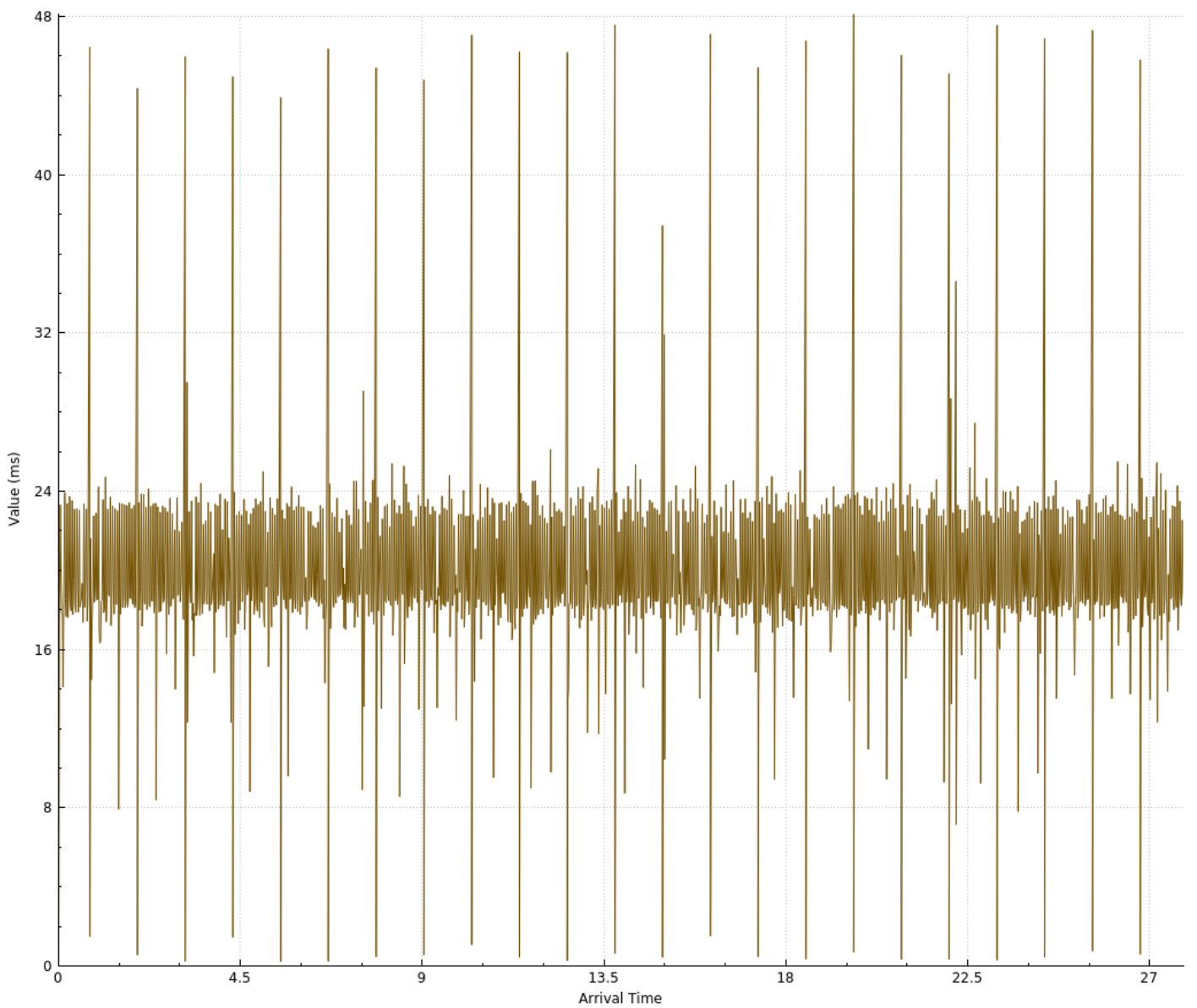
I'm testing the osmo-bts-trx receiver behavior under a variety of conditions.

In case osmo-trx fails to send us some bursts, or in case such bursts are lost, we are seeing a behavior in which the RTP packets on Abis are sent in irregular intervals.

The normal behavior (all bursts from osmo-trx arriving) looks like this:



However, when 8 consecutive burst are dropped at every 10th 26-multiframe, we get:



So whenever the bursts are lost, we get the related RTP packets 44-48ms after the previous one, as opposed to the normal ~20ms

Related issues:

Related to OsmoBTS - Bug #2987: OsmoBTS RxQual/RxLev averaging broken if burs...	Stalled	02/23/2018
Related to OsmoBTS - Bug #2975: OsmoBTS doesn't generate measurement indicati...	Stalled	02/21/2018
Related to OsmoBTS - Bug #3428: Too many contiguous elapsed fn, dropping...	Stalled	07/28/2018

History

#1 - 01/04/2018 10:49 AM - laforge

- Assignee set to pespin

#2 - 01/04/2018 11:10 AM - laforge

- Priority changed from Normal to High

#3 - 02/14/2018 04:26 PM - pespin

Hi Harald,

can you detail how did you test both scenarios? What's the setup difference between both? How did you configure the system to have 8 consecutive burst dropped at every 10th 26-multiframe? Or is it an unwanted behavior?

#4 - 02/14/2018 05:07 PM - laforge

On Wed, Feb 14, 2018 at 04:26:38PM +0000, pespin [REDMINE] wrote:

can you detail how did you test both scenarios? What's the setup difference between both? How did you configure the system to have 8 consecutive burst dropped at every 10th 26-multiframe? Or is it an unwanted behavior?

I have no clue, this follow-up question is unfortunately too long after I did the related work for me to remember the details. It was probably related to implementing some PHY / BER related tests at the time. At least the utils/prbs-tool in the libosmocore.git branch laforge/prbs-tool contains some code like

```
+     /* loose 'count' bursts every 'nth_mframe' on TRX-BTS interface */
+     struct {
+         unsigned int count;
+         unsigned int nth_mframe;
+     } sim_lost_bursts;
```

I used this tool to generate bursts (rather than using osmo-trx) and then fed them into osmo-bts-trx. I remember it was in combination with the fake_trx from fixeria, so basically fake_trx was used to handle the control commands like SETBSIC, etc. and prbs-tool was used to generate the traffic channel bits with configurable errors/drop-outs.

I think I then manually activate some TCH lchans using VTY commands

#5 - 02/15/2018 10:25 AM - fixeria

Hi pespin,

I would like to add my two cents here.

You can actually sniff a normal or virtual connection between a phone and osmo-bts-trx, using a new tool called trx_sniff.py from the FakeTRX toolkit.

Then you can modify the capture, e.g. dropping some bursts, and replay it using another tool called burst_send.py.

The problem is that I still have not finished the implementation of burst_send.py, to make it able to replay the capture files. Anyway, the capture format is described

in burst_send.py. If you would like to use this toolkit,
just let me know and I'll try to finish the implementation ASAP ;)

#6 - 07/18/2018 05:48 PM - pespin

- Status changed from New to In Progress

First analysis steps done. osmo-bts-trx code path when receiving a data payload from osmo-trx up to sending the RTP packet to osmo-mgw:

```
trx_data_read_cb          (osmo-bts-trx)
  trx_sched_ul_burst      (common)
    rx_tchf_fn            (common)
      rx_tchf_fn          (osmo-bts-trx, another for TCH/H)
        _sched_compose_tch_ind (common)
          l1sap_up(SAP_GSM_PH, PRIM_TCH, PRIM_OP_INDICATION) (common)
            l1sap_tch_ind   (common)
```

Some interesting points:

- Pattern is packet N is delayed 20 ms (one packet time), and next packet N+1 then comes with a delta approx = 0, which means there's no time delay degradation in the call (since the one with delta=0 is sent on time).
- Packet N+1 with delta=0 contains most times payload full of zeros (with first byte being 0xd0). The delayed packet N contains what seems to be valid enough data. The zero payload probably comes from "bad frame indicator (bfi) code path tag in scheduler_trx.c:rx_tchf_fn()
- Timestamp and seq looks good for both N and N+1.

This kind of behavior looks like a osmo-bts-trx specific issue, as doesn't match with the code from common part in l1sap_tch_ind(). The only place where this can happen inside osmo-bts code (sending up 2 tch packets pretty quickly) is a while() loop in trx_sched_ul_burst(), where rx_tchf_fn() is called for each "lost" frame numbers once the last one is received, with 0 filled buffers.

TODO:

So all in all, everything looks quite good except that so far I think that the all 0 packet should be the N instead of N+1. And then it should be sent to upper layers in a way in which it is handled correctly by osmo_rtp_skipped_frame() instead of sending a zero payload RTP packet. Hint: msg->len must be zero to be caught by osmo_rtp_skipped_frame()

#7 - 07/19/2018 05:21 PM - pespin

I'm trying to find out the cause of the generated packets with 0 payload, as I don't really understand correctly how they relate to the timings I saw.

I'm trying to reproduce the issue using this patch in osmo-bts-trx to drop the packets, same as you did with prbs-tool:

```

diff --git a/src/osmo-bts-trx/trx_if.c b/src/osmo-bts-trx/trx_if.c
index f3de2453..56d7ded1 100644
--- a/src/osmo-bts-trx/trx_if.c
+++ b/src/osmo-bts-trx/trx_if.c
@@ -513,6 +513,11 @@ rsp_error:
     * TRX burst data socket
     */

+ struct {
+     unsigned int count;
+     unsigned int nth_mframe;
+ } sim_lost_bursts = {.count = 8, .nth_mframe = 10};
+
+ static int trx_data_read_cb(struct osmo_fd *ofd, unsigned int what)
+ {
+     struct trx_llh *llh = ofd->data;
@@ -541,6 +546,21 @@ static int trx_data_read_cb(struct osmo_fd *ofd, unsigned int what)
     rssi = -(int8_t)buf[5];
     toa256 = ((int16_t)(buf[6] << 8) | buf[7]);

+     /* simulate lost frames on TRX <-> BTS interface */
+     if (sim_lost_bursts.count) {
+         /* count number of 26-multiframes */
+         static int count = 0;
+         if (fn % 26 == 0)
+             count++;
+
+         /* every 10th multiframe, drop two entire block of 8 bursts */
+         if ((count % sim_lost_bursts.nth_mframe) == 0 &&
+             (fn % 26) <= sim_lost_bursts.count) {
+             LOGP(DTRX, LOGL_ERROR, "SKIPPING BURST!\n");
+             return 0;
+         }
+     }

+     /* copy and convert bits {254..0} to sbits {-127..127} */
+     for (i = 0; i < burst_len; i++) {
+         if (buf[8 + i] == 255)

```

I finally went the way to use prbs-tool because I only have a LimeSDR-mini right now to use together with osmo-bts-trx, and it's not stable or trustable enough yet for this kind of test.

- 1- I connect fake_trx to osmo-bts-trx
- 2- I activate the correct chan as described in prbs-tool code: bts 0 trx 0 timeslot 2 sub-slot 0 activate fr
- 3- I build prbs-tool from mentioned branch and did a few modifications (like uncommenting the drop parameters in main and commenting the zeroing one).
- 3- Run prbs-tool.

It seems sometimes osmo_ecu_fr_conceal is returning a zeroed payload. In those cases I think we should send an empty payload to l1sap instead to account as an empty RTP packet when sending:

```
20180720123002885 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074369/56/09/1
1/25 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 e
0 08 00 00 30 63 00 00 0a 00 00 00 81 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 dd 3f 64 96 fc 9a 99 80 c6 51 a5 fd 16 3a cb 3c 7d d0 6b 6e c1 6b ea a0 52 bc bb 81 ce 93 d7 51 21
20180720123002895 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074373/56/13/1
5/29 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 e
0 08 00 00 30 63 00 00 0a 00 00 00 85 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 d9 c2 f6 cd 0e f0 ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7
20180720123002913 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074377/56/17/1
9/33 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 e
0 08 00 00 30 63 00 00 0a 00 00 00 89 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 de c9 2d f9 35 33 01 8c a3 4b fa 2c 75 96 78 fb a0 d6 dd 82 d7 d5 40 a5 79 77 03 9d 27 ae a2 43 38
20180720123002936 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/osmo-bts-trx/scheduler_trx.c:1060 074
389/56/03/31/45 (bts=0,trx=0,ts=2) TCH/F: Received bad data (29/104)
20180720123002936 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/osmo-bts-trx/scheduler_trx.c:1092 074
389/56/03/31/45 (bts=0,trx=0,ts=2) TCH/F: osmo_ecu_fr_conceal
20180720123002936 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074382/56/22/2
4/38 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 e
0 08 00 00 30 63 00 00 0a 00 00 00 8e 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 de c9 2d f9 35 33 01 8c a3 4b fa 2c 75 96 78 fb a0 d6 dd 82 d7 d5 40 a5 79 77 03 9d 27 ae a2 43 38
20180720123002955 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/osmo-bts-trx/scheduler_trx.c:1060 074
393/56/07/35/49 (bts=0,trx=0,ts=2) TCH/F: Received bad data (33/104)
20180720123002955 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/osmo-bts-trx/scheduler_trx.c:1092 074
393/56/07/35/49 (bts=0,trx=0,ts=2) TCH/F: osmo_ecu_fr_conceal
20180720123002955 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074386/56/00/2
8/42 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 5
0 0b 00 00 30 63 00 00 0a 00 00 00 92 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20180720123002973 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/osmo-bts-trx/scheduler_trx.c:1060 074
397/56/11/39/01 (bts=0,trx=0,ts=2) TCH/F: Received bad data (37/104)
20180720123002973 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/osmo-bts-trx/scheduler_trx.c:1092 074
397/56/11/39/01 (bts=0,trx=0,ts=2) TCH/F: osmo_ecu_fr_conceal
20180720123002973 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074390/56/04/3
2/46 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 5
0 0b 00 00 30 63 00 00 0a 00 00 00 96 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20180720123002996 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074395/56/09/3
7/51 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 e
0 08 00 00 30 63 00 00 0a 00 00 00 9b 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 d7 e4 d4 cc 04 32 8d 2f e8 b1 d6 59 e3 ee 81 5b 76 09 5f 55 02 15 e4 dc 0e 74 9e ba 89 0c e1 7b 66
20180720123003014 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074399/56/13/4
1/03 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 5
0 0b 00 00 30 63 00 00 0a 00 00 00 9f 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 d8 77 87 fc 1e f8 b9 90 4a 76 8f 3e 6c 54 8e 36 ae 26 22 01 08 c2 72 ac 37 a6 e4 50 ad 3f 64 96 fc
20180720123003033 DL1P <0007> /home/pespin/dev/syscom/git/osmo-bts/src/common/scheduler.c:404 074403/56/17/4
5/07 (bts=0,trx=0,ts=2) : (bts=0,trx=0,ts=2,ss=0) Rx -> RTP: 00 00 00 01 06 00 00 02 00 00 00 00 00 00 00 00 c
0 0d 00 00 30 63 00 00 0a 00 00 00 a3 22 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 d9 a9 98 0c 65 1a 5f d1 63 ac b3 c7 dd 06 b6 ec 16 be aa 05 2b cb b8 1c e9 3d 75
```

#9 - 07/20/2018 11:03 AM - pespin

It seems the zero payload rtp packets are actually valid and refer to FullRate SID indication frames, as specified in TS 101 318 section 5.3.2.

#10 - 07/20/2018 12:09 PM - pespin

- % Done changed from 0 to 70

There seems to be an issue with `osmo_ecu_fr_reset()->conceal_frame()->reduce_xmaxcr_all()`, where if one part of the buffer is silent, then the entire buffer is marked as silent and silenced (zeroed).

After fixing the issue, I don't see the zeroed payloads anymore while testing with the burst-drop scenario described in this task.

Fix submitted in <https://gerrit.osmocom.org/#/c/libosmocore/+10073/>

Regarding time, I didn't experience timing issues so far as described in the original post on this ticket. Analysis of the code didn't provide with any hint regarding this topic. It may have been fixed during last 8 months.

#11 - 07/23/2018 02:02 PM - pespin

After merging <https://gerrit.osmocom.org/#/c/osmo-bts/+10074/>, it was easy to see that `osmo-trx` (both with USRP and limesdr-mini) is not sending all bursts, but only those containing data. That means we cannot rely on current implementation for `trx_sched_ul_burst()` to make sure no burst is dropped since it is expected that it goes through the code path of "elapsed >= 10", where this jump is silenced towards upper layers.

As a result, it can happen that some TCH burst are lost in `trx_sched_ul_burst()` and the RTP clock in `l1sap` becomes out of sync.

In order to fix it, we need to add some code in `trx_sched_ul_burst()` in the case `elapsed >= 10` which calculates the number of lost TCH frames and sends up the stack empty payloads (or zeroed payloads) for those frames lost.

#12 - 07/23/2018 04:14 PM - fixeria

- Related to Bug #2987: *OsmoBTS RxQual/RxLev averaging broken if bursts are missign added*

#13 - 09/21/2018 09:54 AM - pespin

- Related to Bug #2975: *OsmoBTS doesn't generate measurement indications in absence of uplink bursts added*

#14 - 09/21/2018 10:32 AM - pespin

- Related to Bug #3428: *Too many contiguous elapsed fn, dropping... added*

#15 - 09/21/2018 12:21 PM - pespin

For the record:

```
19<fixeria> pespin_: I had some public conversation with Harald, where we concluded that the best option would
  be to send some indications from transceiver if nothing was received...
18<fixeria18> ... nothing was received on Uplink
18<fixeria18> I don't remember all details of that conversation, so I need to have a look again
20<pespin_>30 yes that rings a bell, but I don't remember all the details
20<pespin_>30 that means basically code changes in osmo-trx right?
```

19<fixeria> pespin_: yep, and probably in FakeTRX too, but it would be even more complex I think...
19<fixeria> pespin_: in order to keep the compatibility, I think we should introduce some TRX-CTRL command, which would enable the NwR (nothing was received, or so...) indications
20<pespin_>30 and we should still keep it backward-compatible with current behaviour, since we don't control all TRX apps
18<fixeria18> hehe :)
20<pespin_>30 :) hm not sure if we really need to state that explicitly, we can make it work without that maybe.
20<pespin_>30 we'll see
18<fixeria18> we could just send such indications as regular message on DATA interface but without the burst itself - only header...
18<fixeria18> so, older versions of osmo-bts-trx would complain that 'incorrect message was received due to len != X'
18<fixeria18> this is why we would need to introduce this special command
20<pespin_>30 yes, I was thinking something like that
20<pespin_>30 well that depends on whether the message stops it or not, if it only prints we can accept that I guess
18<fixeria18> but we don't know about the behavior of other applications (outside Osmocom) that are using osmo-bts-trx
20<pespin_>30 hm ok good point. Not sure if there's users other than osmo-bts-trx though.
18<fixeria18> also, we need to make sure that such indications would not be sent earlier than the corresponding UL bursts are received / processed
19<fixeria> pespin_: ok, I've just found this discussion: <https://osmocom.org/issues/3428>

#16 - 10/25/2018 09:37 AM - pespin

- Status changed from In Progress to Closed

Closing, lost burst detection being tracked in other tasks such as [#3428](#)

Files

no_loss.png	96.5 KB	12/02/2017	laforge
8bursts_missing_every_10_mframes.png	179 KB	12/02/2017	laforge
8bursts_missing_every_10_mframes.pcapng	60.3 KB	12/02/2017	laforge