

OsmoBTS - Bug #2987

OsmoBTS RxQual/RxLev averaging broken if bursts are missign

02/23/2018 03:02 PM - laforge

Status: Resolved	Start date: 02/23/2018
Priority: High	Due date:
Assignee: dexter	% Done: 100%
Category: osmo-bts-trx	
Target version:	
Spec Reference:	
Description	
The way how the current measurement processing code is implemented, it simply counts the amount of reported measurements during the measurement period and then builds the average.	
This misses the fact that there should always be a well-defined number of measurements within each measurement period (SACCH multiframe). If there are any measurements missing, it means that bursts were missing, and hence those missing bursts should be substituted with 100% BER. As a result, we want to report the average bit error over all bursts/blocks, not just over those that we have received on L1SAP.	
lower layers may for whatever reason drop bursts/blocks (e.g. CPU overload).	
Related issues:	
Related to OsmoBTS - Bug #2975: OsmoBTS doesn't generate measurement indicati...	Resolved 02/21/2018
Related to OsmoBTS - Bug #2978: OsmoBTS rxlev/rxqual SUB computation complet...	Resolved 02/21/2018
Related to OsmoBTS - Bug #2700: Odd RTP behavior in case of bad / missing spe...	Closed 12/02/2017
Related to OsmoBTS - Bug #3665: TTCN3 BTS_Tests last SACCH burst received too...	Closed 10/23/2018

History

#1 - 02/23/2018 03:03 PM - laforge

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

#2 - 02/23/2018 03:03 PM - laforge

- Related to Bug #2978: OsmoBTS rxlev/rxqual SUB computation completely broken [AMR DTX] added

#3 - 02/25/2018 11:03 PM - neels

Note that the recent handover patches for osmo-bsc fix an issue that sounds just like this, in osmo-bsc. see <http://git.osmocom.org/osmo-bsc/commit?id=83594847a8fa8dee4011000747906f10a09bf652> and maybe <http://git.osmocom.org/osmo-bsc/commit?id=11e31c706663f586e78b531e401de562a07e0785>

#4 - 02/26/2018 08:50 AM - laforge

On Sun, Feb 25, 2018 at 11:03:42PM +0000, neels [REDMINE] wrote:

Note that the recent handover patches for osmo-bsc fix an issue that sounds just like this, in osmo-bsc.

Thanks, but I somehow fail to understand how something in OsmoBSC can fix / work around a problem in OsmoBTS?

If bursts from the PHY are missing in OsmoBTS, we fail to generate correct measurement results, i.e. we still report them to the BSC, but the values in there appear valid but are in fact incorrect (they show a too low BER). So the BSC's computations will be off by whatever difference between the real BER and the BER reported by OsmoBSC.

It's not super-critical as in a "sane" setup, we shouldn't be missing any bursts/blocks from the PHY. If we do this, the system running OsmoBTS is most probably overloaded in terms of CPU resources.

#5 - 02/26/2018 01:04 PM - neels

laforge wrote:

On Sun, Feb 25, 2018 at 11:03:42PM +0000, neels [REDMINE] wrote:

Note that the recent handover patches for osmo-bsc fix an issue that sounds just like this, in osmo-bsc.

Thanks, but I somehow fail to understand how something in OsmoBSC can fix / work around a problem in OsmoBTS?

Not per se of course, just as a reference how the averaging was broken and fixed in the BSC until recently. Maybe the BTS code has similar errors. All I'm saying is, maybe it helps to compare the BSC patch to the BTS code base.

If bursts from the PHY are missing in OsmoBTS, we fail to generate correct measurement results, i.e. we still report them to the BSC, but the values in there appear valid but are in fact incorrect (they show a too low BER). So the BSC's computations will be off by whatever difference between the real BER and the BER reported by OsmoBSC.

Sounds like it is a similar bug as in BSC, where invalid measurement reports were still counted as N, so that dividing by N later gives a too low average.

#6 - 07/04/2018 01:00 PM - laforge

- Assignee changed from sysmocom to dexter

#7 - 07/23/2018 10:12 AM - dexter

I had now a look at the source code and also did some experiments (USRP was not available, so I used OCTBTS). There are a couple of open questions, which I have difficulties with.

My main problem is to understand how many measurements we do expect at all. For a TCH/F this should be 26 and for a TCH/H we should get 13 (?). However it should be a fixed number, which is always the same for each channel type. My experiments with OCTBTS give me measurement counts of 13, 24, 25, 23. At least during the call when I am on TCH/H i would have expected to see 26 measurements.

Also as far as I understand the measurement we get from the mobile always covers an interval of frames. How can we conclude 100%BER if we miss measurement frames?. A missing measurement simply could mean that just the SDCCH frame was missing and the other frames were correct?.

My idea is as follows: When a measurement period done and the RSL measurement result is computed, then we check how many measurements were collected. If measurements are missing we pad them with 100%BER measurements and proceed with the computation.

#8 - 07/23/2018 12:10 PM - laforge

Hi dexter,

On Mon, Jul 23, 2018 at 10:12:54AM +0000, dexter [REDMINE] wrote:

My main problem is to understand how many measurements we do expect at all. For a TCH/F this should be 26 and for a TCH/H we should get 13 (?). However it should be a fixed number, which is always the same for each channel type. My experiments with OCTBTS give me measurement counts of 13, 24, 25, 23. At least during the call when I am on TCH/H i would have expected to see 26 measurements.

The interface between the bts-specific part and the common part works on blocks, not bursts.

A 26-multiframe has 26 slots. 24 are voice codec frames, one of them is empty (for neighbor measurements), and one is SACCH, leaving $25/4=6.25$ blocks per 26-multiframe.

The measurement reporting interval is 102 or 104 frames (depending on TCH vs. SDCCH). That's basically $26*4 = 104$ (TCH) or $51*2 = 102$ (SDCCH).

So during one measurement reporting period on a TCH/F, you will receive

- 4x24 tch bursts (= 24 blocks)
- 4x1 sacch bursts (= 1 block)
i.e. a total of 25 measurements.

For TCH/H, you can do the computation yourself if you look at the definition of TCH/H.

Also as far as I understand the measurement we get from the mobile always covers an interval of frames.

measurement reports (whether uplink or downlink) always cover a measurement reporting period, which is the 102/104 frame period described above.

How can we conclude 100%BER if we miss measurement frames?.

hm?

A missing measurement simply could mean that just the SDCCH frame was missing and the other frames were correct?.

I'm not following you. Are you referring to uplink or downlink measurements? Are you referring to SACCH (/H, /F) or SDCCH as you wrote?

My idea is as follows: When a measurement period done and the RSL measurement result is computed, then we check how many measurements were collected. If measurements are missing we pad them with 100%BER measurements and proceed with the computation.

- we need to make sure the detection of "is a measurement period is done" happens whether or not we received any data from L1, **or** somehow ensure we **always** receive a PH-DATA.ind even if no burst/signal was received.
- if we initialize the buffer/array of uplink measurements with 100% BER instead of mem-setting it to zero, **and** use the **specified** number of blocks/measurements (rather than the actually received count) during the average/computation, then the result should be correct

#9 - 07/23/2018 04:14 PM - fixeria

- Category set to osmo-bts-trx

It seems to be related to the problem, which most likely is the reason of OS#2700. Pay your attention to this part of the scheduler implementation:

<https://git.osmocom.org/osmo-bts/tree/src/common/scheduler.c#n868>

where we calculate the amount of potentially lost ('elapsed') bursts since the last received one. If the amount X is lower than 10, then we compensate the gap by generating X zero-filled dummy (in this context) bursts before forwarding the currently received one. Otherwise, **we just jump** through the lost bursts, and **ignore the gap**.

In scope of measurements, we usually do calculate AVG, so if some amount of bursts is lost, it wouldn't affect the results...

#10 - 07/23/2018 04:14 PM - fixeria

- Related to Bug #2700: Odd RTP behavior in case of bad / missing speech frames added

#11 - 07/24/2018 04:50 AM - laforge

On Mon, Jul 23, 2018 at 04:14:17PM +0000, fixeria [REDMINE] wrote:

In scope of measurements, we usually do calculate AVG, so if some amount of bursts is lost, it wouldn't affect the results...

we also do compute the min/max and standard-deviation since some patches I implemented recently.

#12 - 08/11/2018 01:27 PM - laforge

[dexter](#): Following up on our conversation on Thursday: I just re-read that for TCH/H, a voice block (codec frame) is produced every two blocks, while a FACCH frame will take four blocks, i.e. silence/drop two voice blocks (codec frame).

#13 - 08/13/2018 07:33 AM - dexter

- Status changed from New to In Progress

I just re-read that for TCH/H...

Do you mean that a voice frame consists of 2 TDMA frames (a half-block so to say)? I found that in the osmo-bts code but I was unable to confirm this with the spec.

5 6 7 0 1 2 3 4 (Report
ing interval end)

* For each subchannel we get measurement reports for 2x SDCCH + 1x SACCH

=====
== SDCCH/4 ==
=====

SDCCH = D
SACCH = S
FCCH = F

	1	2	3	4	5	6	7	8	9	0	
0123456789012345678901234567890123456789012345678901234567890123456789012345678901											
DDDD SSSSSSSS				DDDDDDDD	DDDDDDDD	SSSSSSSS			DDDDDDDD	DDDD	(channe l)
3333 22223333				00001111	22223333	00001111			00001111	2222	(subcha nnel)
	2	3							0	1	(Report ing interval end)

* For each subchannel we get measurement reports for 2x SDCCH + 1x SACCH

This means that I should get 3 measurements per user in one interval. When I try it out on osmo-bts this seems to match but I also see intervals that only get back with 2 or even only 1 measurements.

```
Tue Aug 14 18:02:53 2018 <0007> scheduler.c:877 236473/178/03/37/29 Too many contiguous elapsed fn, dropping 5
67
Tue Aug 14 18:02:53 2018 <0007> scheduler.c:877 236565/178/17/27/21 Too many contiguous elapsed fn, dropping 5
12
Tue Aug 14 18:02:53 2018 <0007> scheduler.c:877 236591/178/17/02/47 Too many contiguous elapsed fn, dropping 1
341
Tue Aug 14 18:02:53 2018 <0007> scheduler.c:877 236634/178/08/45/38 Too many contiguous elapsed fn, dropping 1
61
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236668/178/16/28/20 Too many contiguous elapsed fn, dropping 1
03
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2709 (bts=0,trx=0,ts=0,ss=4) Fwd RLL msg CHAN_RQD from LAPDm to A-bis
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2796 (bts=0,trx=0,ts=0,pchan=CCCH+SDCCH4) ss=0 Rx RSL CHAN_ACTIV
Tue Aug 14 18:02:54 2018 <0000> rsl.c:1036 (bts=0,trx=0,ts=0,ss=0): rx Channel Activation in state: NONE.
Tue Aug 14 18:02:54 2018 <0000> rsl.c:1158 chan_nr=0x20 type=0x00 mode=0x00
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:621 Activating SDCCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:621 Activating SACCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:670 Set mode 3, 0, handover 0 on SDCCH/4(0) of trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:735 Set a5/0 uplink for SDCCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:735 Set a5/0 uplink for SACCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:735 Set a5/0 downlink for SDCCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0006> scheduler.c:735 Set a5/0 downlink for SACCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:54 2018 <0000> rsl.c:741 (bts=0,trx=0,ts=0,pchan=CCCH+SDCCH4) (ss=0) SDCCH Tx CHAN ACT ACK
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2739 (bts=0,trx=0,ts=0,ss=0) Rx RSL IMM_ASS_CMD
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236685/178/07/45/37 Too many contiguous elapsed fn, dropping 3
23
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236728/178/24/37/28 Too many contiguous elapsed fn, dropping 6
0
Tue Aug 14 18:02:54 2018 <0004> ll_if.c:515 236728/178/24/37/28 RX L1 frame (bts=0,trx=0,ts=0,ss=0) fn=236728
chan_nr=0x20 MS pwr=16dBm rssi=-21.0 dBFS ber=0.00% (0/456 bits) Ll_ta=0 rqd_ta=0 toa256=20
Tue Aug 14 18:02:54 2018 <0004> measurement.c:297 236728/178/24/37/28 (bts=0,trx=0,ts=0,ss=0) adding measureme
nt (is_sub=1), num_ul_meas=0
Tue Aug 14 18:02:54 2018 <0004> measurement.c:268 (bts=0,trx=0,ts=0,ss=0) meas period end fn:236728, fn_mod:88
, status:1, pchan:CCCH+SDCCH4
Tue Aug 14 18:02:54 2018 <0004> measurement.c:481 (bts=0,trx=0,ts=0,ss=0) Calculating measurement results for
physical channel:CCCH+SDCCH4
Tue Aug 14 18:02:54 2018 <0004> measurement.c:491 (bts=0,trx=0,ts=0,ss=0) Number of UL measurements (1) is les
s than expected (3), will add 2 dummy measurements
Tue Aug 14 18:02:54 2018 <0004> measurement.c:562 (bts=0,trx=0,ts=0,ss=0) Number of SUB measurements (1) is le
ss than expected (3), added 2 dummy SUB measurements
Tue Aug 14 18:02:54 2018 <0004> measurement.c:602 (bts=0,trx=0,ts=0,ss=0) Computed TA256( 6) BER-FULL(66.66%
), RSSI-FULL(- 79dBm), BER-SUB(66.66%), RSSI-SUB(- 79dBm)
Tue Aug 14 18:02:54 2018 <0004> measurement.c:618 (bts=0,trx=0,ts=0,ss=0) UL MEAS RXLEV_FULL(31), RXLEV_SUB(31
),RXQUAL_FULL(7), RXQUAL_SUB(7), num_meas_sub(3), num_ul_meas(3)
Tue Aug 14 18:02:54 2018 <0011> lapd_core.c:920 Store content res. (dl=0x7f75f94ba6d8)
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2709 (bts=0,trx=0,ts=0,ss=0) Fwd RLL msg EST_IND from LAPDm to A-bis
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236779/178/23/37/27 Too many contiguous elapsed fn, dropping 4
8
Tue Aug 14 18:02:54 2018 <0004> ll_if.c:515 236779/178/23/37/27 RX L1 frame (bts=0,trx=0,ts=0,ss=0) fn=236779
```

```

chan_nr=0x20 MS pwr=16dBm rssi=-21.0 dBFS ber=0.00% (0/456 bits) L1_ta=0 rqd_ta=0 toa256=21
Tue Aug 14 18:02:54 2018 <0004> measurement.c:297 236779/178/23/37/27 (bts=0,trx=0,ts=0,ss=0) adding measurement
(is_sub=1), num_ul_meas=0
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236799/178/17/06/47 Too many contiguous elapsed fn, dropping 1
7
Tue Aug 14 18:02:54 2018 <0004> ll_if.c:515 236799/178/17/06/47 RX L1 frame (bts=0,trx=0,ts=0,ss=0) fn=236799
chan_nr=0x20 MS pwr=16dBm rssi=-21.0 dBFS ber=0.00% (0/456 bits) L1_ta=0 rqd_ta=0 toa256=20
Tue Aug 14 18:02:54 2018 <0004> measurement.c:297 236799/178/17/06/47 (bts=0,trx=0,ts=0,ss=0) adding measurement
(is_sub=1), num_ul_meas=1
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2686 (bts=0,trx=0,ts=0,ss=0) Handling RLL msg UNIT_DATA_IND from LAPDm to
MEAS REP
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2596 (bts=0,trx=0,ts=0,ss=0) chan_num:32 Tx MEAS RES valid(2), flags(07)
Tue Aug 14 18:02:54 2018 <0000> rsl.c:2614 (bts=0,trx=0,ts=0,ss=0) Send Meas RES: NUM:0, RXLEV_FULL:31, RXLEV_
SUB:31, RXQUAL_FULL:7, RXQUAL_SUB:7, MS_PWR:56, UL_TA:0, L3_LEN:18, TimingOff:0
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236818/178/10/25/18 Too many contiguous elapsed fn, dropping 1
6
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236830/178/22/37/30 Too many contiguous elapsed fn, dropping 1
2
Tue Aug 14 18:02:54 2018 <0007> scheduler.c:877 236830/178/22/37/30 Too many contiguous elapsed fn, dropping 4
70
Tue Aug 14 18:02:54 2018 <0004> ll_if.c:515 236830/178/22/37/30 RX L1 frame (bts=0,trx=0,ts=0,ss=0) fn=236830
chan_nr=0x20 MS pwr=16dBm rssi=-21.0 dBFS ber=0.00% (0/456 bits) L1_ta=0 rqd_ta=0 toa256=19
Tue Aug 14 18:02:54 2018 <0004> measurement.c:297 236830/178/22/37/30 (bts=0,trx=0,ts=0,ss=0) adding measurement
(is_sub=1), num_ul_meas=2
Tue Aug 14 18:02:54 2018 <0004> measurement.c:268 (bts=0,trx=0,ts=0,ss=0) meas period end fn:236830, fn_mod:88
, status:1, pchan:CCCH+SDCCH4
Tue Aug 14 18:02:54 2018 <0004> measurement.c:481 (bts=0,trx=0,ts=0,ss=0) Calculating measurement results for
physical channel:CCCH+SDCCH4
Tue Aug 14 18:02:54 2018 <0004> measurement.c:500 (bts=0,trx=0,ts=0,ss=0) Number of UL measurements (3)
Tue Aug 14 18:02:54 2018 <0004> measurement.c:569 (bts=0,trx=0,ts=0,ss=0) Number of measurements for SUB (3)
Tue Aug 14 18:02:54 2018 <0004> measurement.c:602 (bts=0,trx=0,ts=0,ss=0) Computed TA256( 20) BER-FULL( 0.00%
), RSSI-FULL(- 21dBm), BER-SUB( 0.00%), RSSI-SUB(- 21dBm)
Tue Aug 14 18:02:54 2018 <0004> measurement.c:618 (bts=0,trx=0,ts=0,ss=0) UL MEAS RXLEV_FULL(63), RXLEV_SUB(63
),RXQUAL_FULL(0), RXQUAL_SUB(0), num_meas_sub(3), num_ul_meas(3)
Tue Aug 14 18:02:55 2018 <0007> scheduler.c:877 236881/178/21/37/29 Too many contiguous elapsed fn, dropping 4
8
Tue Aug 14 18:02:55 2018 <0004> ll_if.c:515 236881/178/21/37/29 RX L1 frame (bts=0,trx=0,ts=0,ss=0) fn=236881
chan_nr=0x20 MS pwr=16dBm rssi=-22.0 dBFS ber=0.00% (0/456 bits) L1_ta=0 rqd_ta=0 toa256=19
Tue Aug 14 18:02:55 2018 <0004> measurement.c:297 236881/178/21/37/29 (bts=0,trx=0,ts=0,ss=0) adding measurement
(is_sub=1), num_ul_meas=0
Tue Aug 14 18:02:55 2018 <0000> rsl.c:2709 (bts=0,trx=0,ts=0,ss=0) Fwd RLL msg DATA_IND from LAPDm to A-bis
Tue Aug 14 18:02:55 2018 <0000> rsl.c:2796 (bts=0,trx=0,ts=0,pchan=CCCH+SDCCH4) ss=0 Rx RSL DEACTIVATE_SACCH
Tue Aug 14 18:02:55 2018 <0006> scheduler.c:621 Deactivating SACCH/4(0) on trx=0 ts=0
Tue Aug 14 18:02:55 2018 <0007> scheduler.c:877 236901/178/15/06/49 Too many contiguous elapsed fn, dropping 1
7
Tue Aug 14 18:02:55 2018 <0007> scheduler.c:877 236932/178/20/37/28 Too many contiguous elapsed fn, dropping 2
8
Tue Aug 14 18:02:55 2018 <0004> ll_if.c:515 236932/178/20/37/28 RX L1 frame (bts=0,trx=0,ts=0,ss=0) fn=236932
chan_nr=0x20 MS pwr=16dBm rssi=-22.0 dBFS ber=0.00% (0/456 bits) L1_ta=0 rqd_ta=0 toa256=19
Tue Aug 14 18:02:55 2018 <0004> measurement.c:297 236932/178/20/37/28 (bts=0,trx=0,ts=0,ss=0) adding measurement
(is_sub=1), num_ul_meas=1
Tue Aug 14 18:02:55 2018 <0004> measurement.c:268 (bts=0,trx=0,ts=0,ss=0) meas period end fn:236932, fn_mod:88
, status:1, pchan:CCCH+SDCCH4
Tue Aug 14 18:02:55 2018 <0004> measurement.c:481 (bts=0,trx=0,ts=0,ss=0) Calculating measurement results for
physical channel:CCCH+SDCCH4
Tue Aug 14 18:02:55 2018 <0004> measurement.c:491 (bts=0,trx=0,ts=0,ss=0) Number of UL measurements (2) is les
s than expected (3), will add 1 dummy measurements
Tue Aug 14 18:02:55 2018 <0004> measurement.c:562 (bts=0,trx=0,ts=0,ss=0) Number of SUB measurements (2) is le
ss than expected (3), added 1 dummy SUB measurements
Tue Aug 14 18:02:55 2018 <0004> measurement.c:602 (bts=0,trx=0,ts=0,ss=0) Computed TA256( 12) BER-FULL(33.33%
), RSSI-FULL(- 51dBm), BER-SUB(33.33%), RSSI-SUB(- 51dBm)
Tue Aug 14 18:02:55 2018 <0004> measurement.c:618 (bts=0,trx=0,ts=0,ss=0) UL MEAS RXLEV_FULL(59), RXLEV_SUB(59
),RXQUAL_FULL(7), RXQUAL_SUB(7), num_meas_sub(3), num_ul_meas(3)
Tue Aug 14 18:02:55 2018 <0011> lapd_core.c:1556 N(S) sequence error: N(S)=0, V(R)=1 (dl=0x7f75f94ba6d8 state
LAPD_STATE_MF_EST)

```

Are there also some DTX rules to apply to SDCCH channels? I think no, the amount of measurement data would be a bit low. We only have 3 blocks there and reducing it even more seems not to be desirable, but I am not sure. So probably those missing blocks are just lost?

I have updated my branch under pmaier/measurement_13082018

#18 - 08/14/2018 05:36 PM - laforge

On Tue, Aug 14, 2018 at 04:52:26PM +0000, dexter [REDMINE] wrote:

This means that I should get 3 measurements per user in one interval.

that is my understanding, too.

When I try it out on osmo-bts this seems to match but I also see intervals that only get back with 2 or even only 1 measurements.

this is odd.

Are there also some DTX rules to apply to SDCCH channels? I think no, the amount of measurement data would be a bit low. We only have 3 blocks there and reducing it even more seems not to be desirable, but I am not sure. So probably those missing blocks are just lost?

I think DTX is not permitted on SDCCH. I think this was explicitly mentioned in one of the related 05.02 / 05.03 or so specs.

#19 - 08/16/2018 09:08 AM - dexter

I have now put the change to Gerrit so we can do some review: <https://gerrit.osmocom.org/#/c/osmo-bts/+10476/> However, there are still some open questions. I am confident that the intervals now work, but the computed results are now different of course.

The TOA256 stuff should yield correct results, but I am not sure. I have changed the unit-tests for that a bit so that it gets full measurements. Otherwise the results would look odd and it is probably not a realistic testcase when there are only three measurement samples.

When we miss a measurement sample, we set `.ta_offs_256bits = 0`. Is this correct? To me it looks like if we were messing up the average with this.

#20 - 08/16/2018 11:30 AM - laforge

On Thu, Aug 16, 2018 at 09:08:27AM +0000, dexter [REDMINE] wrote:

When we miss a measurement sample, we set `.ta_offs_256bits = 0`. Is this correct? To me it looks like if we were messing up the average with this.

That's a good point. Not sure what is "right" here. We simply don't know the timing.

It might make sense to simply exclude those missing bursts from computing the timing min/max/avg. Not sure how easy this is in the current code structure.

#21 - 09/11/2018 07:35 AM - dexter

- % Done changed from 50 to 90

I have checked the status on how `ta_offs_256bits` is currently used in the computation.

In `measurement.c:lchan_meas_check_compute()` one can see that `ta256b_sum += m->ta_offs_256bits`; is only executed when `i < lchan->meas.num_ul_meas`. This means only real `ta_offs_256bits` go into the computation. So from that perspective we should be fine now.

We also decided only to use the `m->inv_rssi` from the real measurements at some point. The code currently computes `ta256b_sum += m->ta_offs_256bits`; `i < lchan->meas.num_ul_meas`. I wonder if this was a good idea since this again messes up the average. If many measurements are missing we still would get a good rssi value. This is probably not right and we should move `ta256b_sum += m->ta_offs_256bits`; a few lines below that it is executed for all computation rounds.

The computation of `ber_full_sum += m->ber10k`; looks good. Here we also include the made up measurements with 100% BER. This should give a realistic average. Also the following code `if (is_sub)...` looks ok to me.

When we are done with the above questions we can close this ticket. There are still problems with the measurement for cases where bursts are missing, but those are handled on another ticket: See also: [#2975](#)

#22 - 10/25/2018 09:40 AM - pespin

- Related to Bug #3665: TTCN3 BTS_Tests last SACCH burst received too late -> wrong fake uplink measurement report added

#23 - 11/05/2018 12:57 PM - dexter

- Status changed from In Progress to Stalled

#24 - 01/20/2020 11:51 AM - dexter

I have revisited that now. No me the calculation of the measurement results looks ok. I found one bug that might occur when no measurements are in buffer. This is tested by the unit-tests but in reality this should not happen.

I wonder whats left to be done. At the moment on recent osmo-bts-trx/osmo-trx versions we can not loose any SACCH frames, that means the computation is triggered reliably, however we can still loose measurement results from the TCH channels but this would be compensated by the current code.

#25 - 05/12/2020 09:56 AM - dexter

- *Status changed from Stalled to Resolved*

- *% Done changed from 90 to 100*

We recently upgraded scheduler.c so that nope indications for TCH channels would also trigger the rx_tchf_fn and rx_tchh_fn functions, so we now also can not loose any bursts for TCH channels as well. I think we are good now and we can close this.