The test verifying measurement results for TCH/H is failing consistently at all time.


What seems odd is that RxQual=7 for the uplink measurement report generated by the BTS. Note this is not the first measurement report after channel activation, but already the second, where we would normally expect all bursts to arrive with perfect quality.

Revision ae07288c - 10/21/2020 02:53 PM - dexter
measurement: count measurements for FACCH/H twice.

The FACCH/H replaces to voice frames in TCH/H. This means during the transmission of a FACCH one measurement value less is handed up to the higher layers. This needs to be compensated by adding the a FACCH/H measurement twice (as if it were from two voice frames).

Change-Id: Id533bb32b7bac40d00d77ba649fe66ea5c0774d
Related: OS#4799

Revision 44fc2523 - 10/26/2020 02:11 PM - dexter
sched_lchan_tchf: count measurements for FACCH/F only once

When the FACCH is generated (while in SPEECH mode), there is also a fake speech indication handed up to 11sap.c. We must make sure that only one of the two indications carry a measurement value, so lets invalidate the measurement values (RSSI in particular) for the generated TCH indication.

Change-Id: le3f2e620ba2a2ab226ddba627fe01c6128f0e0b
Related: OS#4799

Revision 79294134 - 10/26/2020 02:11 PM - dexter
measurement: count all blocks as SUB for TCH/F in signalling mode

SUB frames exist only in voice (or CSD) channels. When a TCH/F is in signalling mode, all blocks must be counted as SUB blocks. (for TCH/H the current implementation is correct.)

Change-Id: l04be21200afa1d03afa0d7e476c66fa79cf42249
Related: OS#4799
measurement: fix expected number of measurements

The logic in measurement.c checks the amount of collected measurement values. This is done for the total amount of measurements and the amount of SUB blocks measurements.

The functions that return the expected number of measurement values currently do not take into account that the mode of a TCH/F or TCH/H has an effect on the number of expected SUB blocks. (In signalling channels all blocks count as SUB). Also a TCH/H in signalling mode generates only half the amount of measurements because the blocks in signalling mode are separated over 6 bursts instead of 4. This also needs to be taken into account.

Change-Id: I01c7b6cc908c647263ab88f6b6281c4732f88779
Related: OS#4799

Revision 85060729 - 10/30/2020 09:45 AM - fixeria
osmo-bts-trx/scheduler: fix comments related to FACCH/H and BFI
Change-Id: I1c1d81ba5a54946089da6c32b0d421411fe71ed8
Related: OS#4799

Revision 484952a3 - 11/02/2020 02:46 PM - dexter
measurement.c: fix integer overflow problem

The variables num_meas_sub_expect - num_meas_sub must not be subtracted without prior checking. Depending on the input (which might be erroneous), num_meas_sub might be greater then num_meas_sub_expect. This eventually leads into odd behavior, which can be difficult to debug.

Change-Id: I381cc637d1c125f279ccf88dbf11460946fe24fe
Related: OS#4799

Revision fd515166 - 03/27/2021 06:37 PM - dexter
sched_lchan_tchh: fix frame number and fill FACCH gap

The measurement results / TCH indications that are handed when decoding the TCH/H are off by two bursts. Since a measurement result / TCH block is expected every two bursts anyway the problem can only be noticed when a FACCH transmission is going on and the frame numbers of the BFI TCH blocks appear to be misaligned towards the FACCH block.

The reason is that the incoming bursts are shifted into a 6 burst wide buffer. The decoding functions always look at the beginning of that buffer while the bursts are shifted into the buffer from the end. A facch will always fit exactly in that buffer but TCH/H blocks are only 4 bursts wide and therefore need two additional bursts until they reach the coverage of the decoding function. Lets fix this by putting the correct frame number (from two bursts before) into the remap functions in order to get the correct beginning of the block.

Since the FACCH transmission uses six blocks it takes out two TCH blocks. This means that if we count the FACCH block we end up with a gap of one TCH voice block. Lets generate a dummy measurement to compensate the gap. This will also match the behavior of the osmo-bts-sysmo phy.

Change-Id: I1ad9fa3815f6eb2b4da608ab7d7f16a87ba1f2f91
Related: OS#4799

History

#1 - 10/12/2020 02:10 PM - dexter
- Status changed from New to In Progress

#2 - 10/13/2020 09:15 PM - dexter
- % Done changed from 0 to 20

I think I have now found out what causes the problem. TC_meas_res_sign_tchh activates the TCH/H as signalling channel SDCCH. This seems to
change the measurement reporting interval.

To my understanding the measurement interval for SDCCH/SACCH would look like:

```
1 1 2 3 4 5 6 7 8 9 0
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

MEASUREMENT REPORTING TCH/H TS4/TS5.0
---------------------------------------------------||---------------------------------------------------
DdDdDdDdDdDdSDdDdDdDdDdDdsDdDdDdDdDdDdSDdDdDdDdDdDdsDdDdDdDdDdDdSDdDdDdDdDdDdsDdDdDdDdDdDdSDdDdDdDdDdDds
M M M M M M M M M M M
A
|______remaps to__________|
```

From this we can expect 13 measurements instead of 25. Since the number of the measurements is included into the rxqual calculation we get values that are far of here. This is why the TTCN3 test fails.

Unfortunately I am not sure if this is correct. In GSM 04.03 I find in chapter 6 Channel configurations that SDCCH + SACCH is a valid configuration. So I think technically this is just a FACCH that is permanently present. I assume that allocating an SDCCH on was tested/used (was it?) in real life setups before, so we can be sure that the signalling data that is decoded from here makes sense.

Given that my assumptions are correct I would adjust the expected number of measurements (this number is used to check if measurements are missing and replaces the missing ones => wrong rxqual) to 13 when the channel is in signalling mode (3).

#3 - 10/13/2020 09:21 PM - fixeria

Hi Philipp,

I think I have now found out what causes the problem. TC_meas_res_sign_tchh activates the TCH/H as signalling channel SDCCH.

I am sorry, but this is incorrect. TCH in signalling mode is still a TCH (i.e. traffic channel), not an SDCCH (i.e. dedicated control channel). The only difference from 'speech' mode is that both sides always send FACCH frames instead of speech frames. The multi-frame layout and thus the measurement reporting interval remain the same.

Best regards,
Vadim.

#4 - 10/14/2020 08:10 AM - laforge

Hi Philipp,

On Tue, Oct 13, 2020 at 09:15:55PM +0000, dexter [REDMINE] wrote:

I think I have now found out what causes the problem. TC_meas_res_sign_tchh activates the TCH/H as signalling channel SDCCH.

this is clearly invalid. The TCH/H must be activated as TCH/H in signaling mode, and nothing else.

#5 - 10/14/2020 06:37 PM - dexter

- File invalid_channel_mode_IE.png added
- % Done changed from 20 to 30

this is clearly invalid. The TCH/H must be activated as TCH/H in signaling mode, and nothing else.

I see, makes also more sense to me. I have fixed that now in the tests:

#6 - 10/14/2020 09:44 PM - dexter

The Fix above will not make the test pass. It just fixes the odd CHANnel ACTIVation message. There is a wired problem in osmo-bts-trx for signalling channels the measurement reporting is definetly broken, for normal TCH/H/FACCH+SACCH usage things should be fine, but this seems to be more by chance as the logic around this is not obvious. I can fix this, but I think we need a bit more test coverage. (there is also still #3780)

I wonder if f_TC_meas_res_periodic could be modified so that it tests real TCH/H and not only the signalling case. For sure it is easily possible to activate the channel as TCH/H in GSM V1 speech mode for example but this still does not change the fact that only FACCH (L1CTL_DATA_IND) are
sent by the testcase (are they really or is this generated by TRXCON because the testcase does not send any traffic). I would need L1CTL_TRAFFIC_IND to be send to TRXCON which contain GSM V1 voice frames.

There is this f_TC_meas_res_periodic() that uses as_l1_dcch(), I tried to make an as_l1_tch()

```c
private altstep as_l1_tch() runs on ConnHdlr {
    var L1ctlDlMessage l1_dl;
    [] L1CTL.receive(tr_L1CTL_TRAFFIC_IND(g_chan_nr, tr_RslLinkID_DCCH(?))) -> value l1_dl {
        log("TCH received: ", l1_dl.payload.traffic_ind.data);
        var octetstring pl := '010301'O;
        log("=====================> I AM SENDING TO L1CTL:");
        log(ts_L1CTL_TRAFFIC_REQ(g_chan_nr, ts_RslLinkID_DCCH(0), f_pad_oct(pl, 23, '2B'O)));
        L1CTL.send(ts_L1CTL_TRAFFIC_REQ(g_chan_nr, ts_RslLinkID_DCCH(0), f_pad_oct(pl, 23, '2B'O)));
    }
    repeat;
}
```

But that did not work, which might be because there is never an tr_L1CTL_TRAFFIC_IND coming from TRXCON, shouldn't I receive something as soon as the channel is activated?

#7 - 10/18/2020 07:49 PM - fixeria

Hi Philipp,

The Fix above will not make the test pass. It just fixes the odd CHANnel ACTIVation message.

I am wondering whether this 'Channel Rate and Type' IE is even considered in osmo-bts...

There is a wired problem in osmo-bts-trx for signalling channels the measurement reporting is definetly broken, for normal TCH/H/FACCH+SACCH usage things should be fine,

For measurement reporting on TCH/F or TCH/H there is no difference whether the channel is activated in signalling or in traffic mode - TCH is a separate channel, SACCH is a separate channel. Why do you think it should be fine for traffic?

I wonder if f_TC_meas_res_periodic could be modified so that it tests real TCH/H and not only the signalling case.

I don't think the 'traffic' mode would change anything (see above).

```c
[[ L1CTL.receive(tr_L1CTL_TRAFFIC_IND(g_chan_nr, tr_RslLinkID_DCCH(?))) -> value l1_dl
    [...] shouldn't I receive something as soon as the channel is activated?
```

The problem is that you're not sending any RTP frames to the BTS, so you should see lots of:

```
DL1P INFO sched_lchan_tchf.c:539 001477/01/21/49/41 (bts=0,trx=0,ts=1) TCH/F: No TCH or FACCH prim for transmi

t.
DL1P INFO sched_lchan_tchf.c:539 001482/01/00/03/46 (bts=0,trx=0,ts=1) TCH/F: No TCH or FACCH prim for transmi

t.
DL1P INFO sched_lchan_tchf.c:539 001486/01/04/07/50 (bts=0,trx=0,ts=1) TCH/F: No TCH or FACCH prim for transmi

t.
DL1P INFO sched_lchan_tchf.c:539 001490/01/08/11/02 (bts=0,trx=0,ts=1) TCH/F: No TCH or FACCH prim for transmi

t.
```

Most likely, it's just sending dummy bursts (I would expect BFIs instead?), so trxcon fails to decode TCH/F blocks:

```
202010019024408686 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1788 for TCH/F
202010019024408704 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1792 for TCH/F
202010019024408727 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1797 for TCH/F
202010019024408746 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1801 for TCH/F
202010019024408764 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1805 for TCH/F
202010019024408787 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1810 for TCH/F
202010019024408806 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1814 for TCH/F
202010019024408824 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1818 for TCH/F
202010019024408847 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1823 for TCH/F
202010019024408866 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1827 for TCH/F
202010019024408884 DSCHD ERROR sched_lchan_tchf.c:127 Received bad TCH frame ending at fn=1831 for TCH/F
```

05/16/2021
The problem is that you're not sending any RTP frames to the BTS, so you should see lots of:

this still applies, but regardless of that trxcon must be sending BFIs to the test case. I did a quick investigation, and found out that the test suite does not send the correct TCH mode to trxcon, so trxcon always works in signalling mode and sends empty DATA.ind (indicating bad xCCCH frames) instead. Moreover, I found out that the 'L1ctlDmEstReq' definition in L1CTL_Types is incorrect:

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20753 library/L1CTL_PortType: fix indention in alt() statements [NEW]
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20754 library/L1CTL_Types: turn L1ctlTchMode into an enumerated type [NEW]
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20755 library/L1CTL_Types: fix definition of L1ctlDmEstReq [NEW]

I don't think the 'traffic' mode would change anything (see above).

I submitted an incomplete patch set (still need to fix some things):

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20756 BTS_Tests: cosmetic: rename 'as_l1_dcch' to 'as_l1_dcch_loop' [NEW]
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20757 BTS_Tests: introduce and use TCH loop - as_l1_tch_loop() [NEW]
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20758 BTS_Tests: introduce TC_meas_res_speech_tch{f,h} [NEW]

and indeed, you were right: both TC_meas_res_speech_tchf and TC_meas_res_speech_tchh pass. I am surprised. Does it mean that the measurement processing logic in osmo-bts-trx is somehow wrong?

Best regards,
Vadim.

#9 - 10/19/2020 07:50 PM - fixeria

- Subject changed from TC_meas_res_sign_tchh failing to Both TC_meas_res_sign_tchh and TC_meas_res_sign_tchh_toa256 are failing

I also had a quick look at TC_meas_res_sign_tchh_toa256(), which currently aborts due to a DTE:


with this patch applied, it still does not pass, but it's 'speech' variant does:


#10 - 10/22/2020 04:42 PM - dexter

I have found out that the remaining reasons why TC_meas_res_sign_tchh_ are failing is a mishandling of the FACCH in sched_lchan_tchh.c. The signalling mode can be seen as a permanent FACCH transmission but without TCH indications (l1sap.c). Since the FACCH is spreaded over 6 bursts the amount of measurement values is halved. In order to correct the amount we could add each measurement we get from a FACCH twice. This is for sure debatable, but it corrects the amount of measurements. See also:


For FACCH/F we have the opposite issue. Here it is actually simple to correct the problem. We just make sure that the TCH/F indication that is generated as a replacement for the missing speech block. Since FACCH/F blocks are exactly the same size as speech blocks everything matches up again. See also:

https://gerrit.osmocom.org/c/osmo-bts/+/20841 sched_lchan_tchf: count measurements for FACCH/F only once

In order to have a testbed for the FACCH debugging I have added tests that open the channel in speech mode while occasionally injecting FACCH blocks. For TCH/F everything looks fine here, but even when the test passes for TCH/H it still looks very messy from inside of osmo-bts. Unfortunately there is no easy way to check if osmo-bts gets the right amount of upling measurements. A rate counter that is queried from TTCN3 might help here.


So far now all tests should pass, even when I am not happy with the result. I have studied the specs and the FACCH behavior of osmo-bts but unfortunately I can not align the theory with the actual implementation. This was also the case when I was doing the AMR DTX implementation. However, here is the theory I currently know:
**FACCH/F**

---

**FACCH = F**

**SACCH = S**

---

**MEASUREMENT REPORTING TCH/F TS1 (SIGNALLING)**

---

**MEASUREMENT REPORTING TCH/F TS1 (SPEECH)**

---

**Example: FACCH/F**

---

**FACCH/H**

---

**FACCH = F for subchannel 0**

**FACCH = f for subchannel 1**

**SACCH = S**

---

**MEASUREMENT REPORTING TCH/H TS4/TS5.0 (SIGNALLING)**

---

**MEASUREMENT REPORTING TCH/H TS4/TS5.0 (SPEECH)**

---

**Example: FACCH/H on SS0**

---

05/16/2021
Note: The FACCH gets the same bandwidth (8 half bursts) as in TCH/F, in the middle of the transmission the bandwidth doubles so only 6 bursts are needed. In this scheme, the FACCH replaces two TCH/H speech frames, which also means that one block less is generated for higher layers. This eventually leads also to one measurement result less.

What confuses me the most in osmo-bts is that the ongoing_facch flag is set after the FACCH is generated. I would expect this condition somewhere in between. But maybe I am overlooking something else.

There also a completely different question that came up: Is it even allowed to replace a SUB frame with a FACCH?

However, maybe its better to review the patches first to see what makes sense here and how it can be improved. It would also good to know if the above is correct. (important: the "M" markers indicate the location of the frame number from the beginning of the frame.)

#11 - 10/23/2020 08:10 PM - fixeria
- File FACCH_H-1.jpg added

Hi Philipp,

as promised, I am attaching my old paintings that I used while working on TCH/H support for trxcon. They're quite draftish and inaccurate, but should shed some light on how FACCH/H decoding is done in both osmo-bts-trx and trxcon. The problem is that in osmo-bts-trx we're still sending only one BFI (see step 'f'). An additional BFI needs to be sent (in speech mode) when the buffer contains a complete FACCH/H frame (see steps 'c' or 'd'). This way we keep the frame rate: one frame on each 2nd received burst. I can write a patch quickly.

Kind regards,
Vadim.

#12 - 10/23/2020 08:12 PM - dexter
- % Done changed from 40 to 60

fixeria and me have discussed some of the problems of the TCH/H and the measurement processing (thanks for the scan of your notes!) We decided not to use the approach proposed in #20840 (count FACCH blocks twice.) We think that measurement.c should detect and handle measurements coming from FACCH.

The patch that prevents measurement (RSSI) values attached to the BFI (fake TCH) indication is still in review:
https://gerrit.osmocom.org/c/osmo-bts/+/20841 sched_lchan_tchf: count measurements for FACCH/F only once

Then there was a small bug in the SUB frame detection logic, which I corrected now:
https://gerrit.osmocom.org/c/osmo-bts/+/20852 measurement: count all blocks as SUB for TCH/F in signalling mode

Since the amount of measuremnt values and sub frames is indeed different depending on the channel mode, I fixed this as well.
https://gerrit.osmocom.org/c/osmo-bts/+/20853 measurement: fix expected number of measurements

Not yet fixed is the problem we have when a FACCH is injected into an TCH/H channel. If this happens, and a sub frame is hit the results get messed up. This is why TC_meas_res_speech_tchh_facch does not yet pass. I still wonder if it is even legal to take out a voice frame for a FACCH frame. I wouldn't be surprised if this would be forbidden.

I would now do the following: If a FACCH is received, the two BFI indications will not carry RSSI values and because of this only the FACCH measurement value is in the list. When the measurement logic counts out the measurements it will detect that there were FACCH frames and make sure that this taken into account properly (expect less sub measurements, expect less measurements in total, don't compensate missing measurements.)
Sorting out the remaining problems with TC_meas_res_speech_tchh_facch is actually not that easy. My original idea was to add code to measurement.c that accounts the FACCH measurement samples differently so that the one less measurement sample that is generated when the FACCH is transmitted is taken into account when doing the measurement calculations. Unfortunately this did not work out.

(I am still not getting smart out of the frame numbers that mark the beginning of the FACCH. The FACCH has the same frame number as one of the BFI TCH samples. This does not match the theory, I will check with sysmobts tomorrow to see if the frame numbers look different there.)

See also: https://gerrit.osmocom.org/c/osmo-bts/+/20980

fixeria had a nice idea, which I probably did not understand correctly, but I think we are on the right way here. If we are in SPEECH mode, we will not use the FACCH to get the measurement samples. Instead we will only use the generated TCH indications. This way the stream of measurement samples is evened out. For TCH/F it is not a problem anyway because FACCH and voice blocks have the same length. The problem only manifests itself in TCH/H, where FACCH and voice blocks have different length. With the current version of my patch TC_meas_res_speech_tchh_facch is passing.

See also: https://gerrit.osmocom.org/c/osmo-bts/+/21024

#14 - 11/23/2020 04:03 PM - dexter

- File osmo_bts_trx_tch_f_facch.log added
- File osmo_bts_trx_tch_h_facch.log added
- File sysmo_bts_trx_tch_f_facch.log added
- File sysmo_bts_trx_tch_h_facch.log added

I have now analyzed and compared with the behavior we see with osmo-bts-sysmo. With osmo-bts-sysmo I find the same behavior for TCH/F as we currently have it in osmo-bts-trx, but on TCH/H the behavior is different. On osmo-bts-sysmo, the phy seems to mask out the one missing TCH measurement, which seems logical to me because it makes the calculation much easier. For osmo-bts-trx the behavior looks completely messed up (see report below).

Behavior of the sysmobts-phy

---

== FACH/F ==
---

---8<-------- sysmo_bts_trx_tch_f_facch.log --------8<---
---

<0000> rsl.c:3003 (bts=0,trx=0,ts=1,ss=0) Fwd RLL msg EST_IND from LAPDm to A-bis
<0000> rsl.c:3098 (bts=0,trx=0,ts=1,ss=0) Fwd RLL msg DATA_IND from LAPDm to A-bis
<0000> rsl.c:3098 (bts=0,trx=0,ts=1,ss=0) Rxs recalculating sync params

---8<-------- sysmo_bts_trx_tch_h_facch.log --------8<---
---

TCH = T
FACCH = F (Half FACCH, Half TCH/VOICE)
SACCH = S

Possible slots to transmit a FACCH (same for uplink and downlink)

M-------M--- ----M------- M-------M--- ----M------- M-------M--- ----M------- M-------M--- ----M------- M-------M---

=> Theory matches observation. There is no BFI indication whatsoever for the lost TCH block. The FACCH block replaces the TCH block. Measurement results come exactly as if there would be no FACCH occurrence.

----------8<---------- sysmo_bts_tch_f_facch.log ----------8<----------

---------8<--------- sysmo_bts_tch_f_facch.log --------8<--------
scheduler.c:1025: Deactivating SACCH/4(0) on trx=0 ts=0

rsl.c:3003: (bts=0, trx=0, ts=1, ss=0) Fwd RLL msg DATA_IND from LAPDm to A-bis

measurement.c:347: 061646/46/00/38/50 (bts=0, trx=0, ts=1, ss=0) adding measurement

===================> GOT MEAS: FACCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=0, inv_rssi=55, fn=78

measurement.c:347: 061641/46/21/33/45 (bts=0, trx=0, ts=1, ss=0) adding measurement

measurement.c:347: 061645/46/25/37/49 (bts=0, trx=0, ts=0, ss=0) adding measurement

measurement.c:347: 061641/46/21/33/45 (bts=0, trx=0, ts=1, ss=0) adding measurement

behavior of the osmo-btx-trx

FACCH/F

---------8<-------- osmo_bts_trx_tch_f_facch.log --------8<---------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=98, fn=21

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=5026, inv_rssi=55, fn=30

GOT MEAS: FACCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=5026, inv_rssi=55, fn=30

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=98, fn=21

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=5026, inv_rssi=55, fn=30

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=96, fn=21

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=5026, inv_rssi=55, fn=30

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26

---8<-------- osmo_bts_trx_tch_f_facch.log ---8<--------

GOT MEAS: TCH, lchan=(bts=0, trx=0, ts=1, ss=0), ber10k=10000, inv_rssi=99, fn=26
Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 061650/46/04/42/02 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=0, ta_offs=-13, ci=0.00, is_sub=0, rssi=-55), num_ul_meas=36, fn_mod=82

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 061650/46/04/42/02 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=0, ta_offs=-13, ci=0.00, is_sub=0, rssi=-55), num_ul_meas=36, fn_mod=82

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 061654/46/08/46/06 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=0, ta_offs=-13, ci=0.00, is_sub=0, rssi=-55), num_ul_meas=37, fn_mod=86

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 061659/46/13/00/11 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=0, ta_offs=-13, ci=0.00, is_sub=0, rssi=-55), num_ul_meas=38, fn_mod=91

---------8<--------- osmo_bts_trx_tch_h_facch.log ---------8<---------

== FACCH/H ==

with the osmo-bts phy.

exactly as if there would be no FACCH occurrence. Behavior matches exactly what we also can see

fn=82: Measurement from TCH after FACCH (overlaps half with second FACCH)
fn=78: Measurement from second FACCH (overlaps half with first FACCH and adjacent TCH)
fn=73: Measurement from first FACCH (overlaps half with second FACCH and adjacent TCH)
fn=69: Measurement from TCH before FACCH (overlaps half with the first FACCH)
fn=64: Measurement from TCH (overlaps with adjacent FACCH)
fn=39: Measurement from TCH (overlaps with adjacent FACCH)
fn=30: Measurement from TCH before FACCH (overlaps half with the FACCH)

Possible slots to transmit a FACCH (same for uplink and downlink)

\[ \begin{array}{cccccccccccc}
\end{array} \]

fn=30: Measurement from TCH before FACCH (overlaps half with the FACCH)
fn=34: Measurement from FACCH (overlaps with adjacent TCH)
fn=39: Measurement from TCH after FACCH (overlaps half with the FACCH)

fn=69: Measurement from TCH before FACCH (overlaps half with the first FACCH)
fn=73: Measurement from first FACCH (overlaps half with second FACCH and adjacent TCH)
fn=78: Measurement from second FACCH (overlaps half with first FACCH and adjacent TCH)
fn=82: Measurement from TCH after FACCH (overlaps half with second FACCH)

--> Theory matches observation. The FACCH block replaces the TCH block. Measurement results come exactly as if there would be no FACCH occurrence. Behavior matches exactly what we also can see with the osmo-bts phy.

-- FACCH/H --

---------8<--------- osmo_bts_trx_tch_h_facch.log ---------8<---------

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 032539/24/13/01/27 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=0, ta_offs=-13, ci=0.00, is_sub=0, rssi=-55), num_ul_meas=39, fn_mod=95

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 032547/24/21/09/35 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=0, ta_offs=-13, ci=0.00, is_sub=0, rssi=-55), num_ul_meas=38, fn_mod=91

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 032550/24/13/27/01 (bts=0,trx=0,ts=1,ss=0) adding measurements t (ber10k=10000, ta_offs=0, ci=0.00, is_sub=1, rssi=-97), num_ul_meas=0, fn_mod=37

Mon Nov 23 15:42:23 2020 <0004> measurement.c:347 032573/24/21/35/09 (bts=0,trx=0,ts=1,ss=0) adding measurements

05/16/2021 11/16
There is good news. I had another look at the problem and this time I think I found the root cause:

-------------
= FACCH/H  =
-------------

It seems that the measurements for the FACCH blocks do not match the expected frame numbers. The theory and observation do not align, and it is possible that the measurements are missing or incorrect.

---

There is good news. I had another look at the problem and this time I think I found the root cause:

-------------
= FACCH/H  =
-------------

It seems that the measurements for the FACCH blocks do not match the expected frame numbers. The theory and observation do not align, and it is possible that the measurements are missing or incorrect.
GOT MEAS: TCH, lchan=(bts=0,trx=0,ts=5,ss=0), ber10k=0, inv_rssi=100, fn=56
Mon Nov 23 22:04:10 2020 DMEAS <0004> measurement.c:347 001616/01/04/35/28 (bts=0,trx=0,ts=5,ss=0) adding measurement (ber10k=0, ta_offs=512, ci=0.00, is_sub=1, rssi=-100), num_ul_meas=5, fn_mod=56

TCH UP, fn=67, fn_begin=60

GOT MEAS: TCH, lchan=(bts=0,trx=0,ts=5,ss=0), ber10k=0, inv_rssi=100, fn=56
Mon Nov 23 22:04:10 2020 DMEAS <0004> measurement.c:347 001620/01/08/39/32 (bts=0,trx=0,ts=5,ss=0) adding measurement (ber10k=0, ta_offs=512, ci=0.00, is_sub=0, rssi=-100), num_ul_meas=6, fn_mod=60

TCH UP, fn=71, fn_begin=65

GOT MEAS: TCH, lchan=(bts=0,trx=0,ts=5,ss=0), ber10k=0, inv_rssi=100, fn=60
Mon Nov 23 22:04:10 2020 DMEAS <0004> measurement.c:347 001625/01/13/44/37 (bts=0,trx=0,ts=5,ss=0) adding measurement (ber10k=0, ta_offs=512, ci=0.00, is_sub=0, rssi=-100), num_ul_meas=7, fn_mod=60

TCH UP, fn=75, fn_begin=69

GOT MEAS: TCH, lchan=(bts=0,trx=0,ts=5,ss=0), ber10k=0, inv_rssi=100, fn=65
Mon Nov 23 22:04:10 2020 DMEAS <0004> measurement.c:347 001629/01/17/48/41 (bts=0,trx=0,ts=5,ss=0) adding measurement (ber10k=0, ta_offs=512, ci=0.00, is_sub=0, rssi=-100), num_ul_meas=8, fn_mod=65

TCH UP, fn=80, fn_begin=73

GOT MEAS: TCH/BFI fn=82

TCH UP (SUPRESSED), fn=88, fn_begin=82

GOT MEAS: FACCH, lchan=(bts=0,trx=0,ts=5,ss=0), ber10k=0, inv_rssi=100, fn=82

FACCH UP, fn=88, fn_begin=82

GOT MEAS: FACCH fn=88

TCH UP, fn=93, fn_begin=86

RESTORE!

GOT MEAS: TCH, lchan=(bts=0,trx=0,ts=5,ss=0), ber10k=0, inv_rssi=100, fn=86

Mon Nov 23 22:04:10 2020 DMEAS <0004> measurement.c:347 001646/01/08/14/06 (bts=0,trx=0,ts=5,ss=0) adding measurement (ber10k=0, ta_offs=512, ci=0.00, is_sub=0, rssi=-100), num_ul_meas=12, fn_mod=86

Note: We ignore measurement result at fn_begin=82, fn=88 as it is has been added artificially to test something.
Assume that FACCH is in the right position, pull other measurement results back by two...

Now transform back, add FACCH from above:

\[ \text{tts\_STT\ tet\_SFF\ FFs\_SFF\ FFs\_STT\ tts\_STT} \]

If we erase the '?' measurement report everything matches up the theory. We see two erased measurements in the results at the positions where they should be.

The function \( rx\_tchh\_fn \) uses a shift register buffer scheme that is 6 bursts deep. The problem is presumably that the decoding functions look at the beginning of the buffer but incoming bursts are added at the end. This explains why the FACCH is decoded in the right position, it fits exactly.

The TCH blocks are always decoded with a two burst position delay. If we take that into account we can fix the frame number problem.

I also tried a hack that fills the burst gaps with measurements, so that the it looks just like with osmo-bts-sysmo. Then BTS_Tests.TC_meas_res_speech_tchh_facch also passes.

---

I have now fixed the problem with the oddly aligned frame numbers by feeding the frame number of the actual block end into the remap functions that calculate \( fn\_begin \). This means I use the frame number from two bursts before here.

I also managed to fill the resulting gap at the correct position. The behavior is now almost identical to the that what I can observe at osmo-bts-sysmo, however there seem to be no BFI indications (RSSI=0) handed up. The testcase BTS_Tests.TC_meas_res_speech_tchh_facch now passes nicely and the frame numbers look good.

See also: [https://gerrit.osmocom.org/c/osmo-bts/+/21329](https://gerrit.osmocom.org/c/osmo-bts/+/21329) sched_lchan_tchh: fix frame number and fill FACCH gap

---

I have fixed the remaining problems in #21329, the patch now calculates the frame numbers directly and does not need another shift buffer to memorize the frame number. This is also more reliable since on the channel activation the shift buffer would not contain the correct frame numbers. However, I still use memset/memcpy twice, which could be optimized out if necessary. I do not know how critical that is, it also only happens on FACCH transmissions.

See also: [https://gerrit.osmocom.org/c/osmo-bts/+/21329](https://gerrit.osmocom.org/c/osmo-bts/+/21329)

---

Both test cases still fail, and we still have not reached an agreement on how to proceed.

---

I have rebased [https://gerrit.osmocom.org/c/osmo-bts/+/21329](https://gerrit.osmocom.org/c/osmo-bts/+/21329) to current master. Looking at the tests I see that TC_meas_res_speech_tchh_facch is failing, which is expected. Unfortunately I am running out of ideas how to fix this.

---

Both test cases still fail, and we still have not reached an agreement on how to proceed.

---

I have rebased [https://gerrit.osmocom.org/c/osmo-bts/+/21329](https://gerrit.osmocom.org/c/osmo-bts/+/21329) to current master. Looking at the tests I see that TC_meas_res_speech_tchh_facch is failing, which is expected. Unfortunately I am running out of ideas how to fix this.

---

Both test cases still fail, and we still have not reached an agreement on how to proceed.
I will work on this once I have time.

#22 - 03/24/2021 04:11 PM - fixeria

- Subject changed from Both TC_meas_res_sign_tchh and TC_meas_res_sign_tchh_toa256 are failing to BTS_Tests.TC_meas_res_speech_tchh_facch is failing

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_channel_mode_IE.png</td>
<td>81.2 KB</td>
<td>10/14/2020</td>
<td>dexter</td>
</tr>
<tr>
<td>FACCH_H-1.jpg</td>
<td>810 KB</td>
<td>10/23/2020</td>
<td>fixeria</td>
</tr>
<tr>
<td>osmo_bts_trx_tch_f_facch.log</td>
<td>135 KB</td>
<td>11/23/2020</td>
<td>dexter</td>
</tr>
<tr>
<td>osmo_bts_trx_tch_h_facch.log</td>
<td>418 KB</td>
<td>11/23/2020</td>
<td>dexter</td>
</tr>
<tr>
<td>sysmo_bts_tch_f_facch.log</td>
<td>43.1 KB</td>
<td>11/23/2020</td>
<td>dexter</td>
</tr>
<tr>
<td>sysmo_bts_tch_h_facch.log</td>
<td>89.6 KB</td>
<td>11/23/2020</td>
<td>dexter</td>
</tr>
</tbody>
</table>