OsmoBSC - Bug #3716

Use SACCH for MO/MT SMS signalling during a voice call

11/29/2018 11:47 PM - fixeria

Status:	Resolved	**Start date:**	11/30/2018
Priority:	Low	**Due date:**	
Assignee:	fixeria	**% Done:**	100%
Category:	A interface		
Target version:			
Spec Reference:			

Description

It seems to be normal to use SACCH instead of FACCH for SMS during a voice call.
In this case every second Measurement Report is replaced by GSM 04.11 messages.

It is supported in OpenBSC, because the information about RAN connection is easily available there, but OsmoMSC has no access to this information (yet?), so FACCH is used.

See paging_cb_mmsms_est_req() in 'src/libmsc/gsm_04_11.c':

```
/* FIXME: specify SACCH in case we already have active TCH */
trans->dlci = 0x03;
```

We should detect somehow if a subscriber has an active TCH connection, and use SACCH.
We can also make this feature configurable from the VTY.

History

#1 - 11/30/2018 07:30 AM - laforge

On Thu, Nov 29, 2018 at 11:47:38PM +0000, fixeria [REDMINE] wrote:

> It seems to be normal to use SACCH instead of FACCH for SMS during a voice call.

I think it's even a clear requirement in the spec to do so.

> It is supported in OpenBSC, because the information about RAN connection is easily
> available there, but OsmoMSC has no access to this information (yet?), so FACCH is used.
>
> See paging_cb_mmsms_est_req() in 'src/libmsc/gsm_04_11.c':
>
> ```
> > /* FIXME: specify SACCH in case we already have active TCH */
> > trans->dlci = 0x03;
> >
> ```
>
> We should detect somehow if a subscriber has an active TCH connection, and use SACCH.

Actually, I think it would be rather odd if we even went through the paging code if we know there's already a connection.

In fact, shouldn't gsm411_mmsms_est_req() detect that there's already asubscr_conn, and only ever call subscr_request_conn() and hence trigger a paging request if there is no pre-existing connection?

We currently check for "if (trans->conn != NULL)", which is true if gsm411_alloc_mt_trans() detects the subscriber already has an active connection. When gsm411_mmsms_est_req() is then executed, we should go into the "if (trans->conn != NULL)" clause and never end up hitting the code path you describe?

> We can also make this feature configurable from the VTY.

no. We should always send any SMS over SAPI3 on the SACCH if a TCH is active. Please note that prioritization of SAPI 0 (signalling) over SAPI 3 (SMS) is also important to handle correctly here. Not sure if we do that properly.

#2 - 11/30/2018 10:59 AM - fixeria

- Status changed from New to Feedback

- Assignee set to fixeria

> I think it's even a clear requirement in the spec to do so.
> We should always send any SMS over SAPI3 on the SACCH if a TCH is active.

Ok, I found it: GSM TS 04.11, sections 2.2-2.3.

> Actually, I think it would be rather odd if we even went through
> the paging code if we know there's already a connection.

But what if BSC would allocate a TCH channel (e.g. due to SDCCH congestion)?
Should we consider this case?

> Please note that prioritization of SAPI 0 (signalling) over SAPI 3 (SMS)
> is also important to handle correctly here. Not sure if we do that properly.

Where should I look to verify this?

Thanks for your notes!

#3 - 11/30/2018 12:55 PM - fixeria

The situation is even more complicated than I initially thought.

First of all, I've introduced a regression during the recent refactoring.
Here is a fix: https://gerrit.osmocom.org/#/c/osmo-msc/+/12043/

Also, I think this is a task of the BSC/BTS to decide, which lchan to use,
i.e. SDCCH or SACCH. Why should the MSC care about that?

Finally, I just tested a MT SMS transfer (initiated from the VTY) during
a voice call between two (not virtual, physical) phones with the fix applied.

Please check out the BSSAP/RSL/LAPDm capture attached:

Frame #76: (BSSAP, **SAPI=0x03**) MSC initiates MT SMS transfer (CP-/RP-DATA)
Frame #78: (RSL, **SAPI=0x00**, C-bits: Bm + ACCH) BSC forwards CP-/RP-DATA to the BTS
Frame #83: (LAPDm, **SAPI=0x00**, FACCH/F) BTS transmits 2nd (final) fragment of CP-/RP-DATA to the MS

Frame #96: (LAPDm, SAPI=0x03, SACCH/F) MS responds with CP-ACK to the BTS
Frame #97: (RSL, SAPI=0x03, C-bits: Bm + ACCH) BTS forwards CP-ACK to the BSC
Frame #98: (BSSAP, SAPI=0x03) BSC forwards CP-ACK to the MSC

Frame #123: (LAPDm, SAPI=0x03, SACCH/F) MS responds with CP-DATA/RP-ACK to the BTS
Frame #124: (RSL, SAPI=0x03, C-bits: Bm + ACCH) BTS forwards CP-DATA/RP-ACK to the BSC
Frame #125: (BSSAP, SAPI=0x03) BSC forwards CP-DATA/RP-ACK to the MSC

Frame #127: (BSSAP, **SAPI=0x03**) MSC responds with CP-ACK
Frame #129: (RSL, **SAPI=0x00**, C-bits: Bm + ACCH) BSC forwards CP-ACK to the BTS
Frame #130: (LAPDm, **SAPI=0x00**, FACCH/F) BTS transmits CP-ACK to the MS

As you can see, during MT message transfer (MSC -> BSC -> BTS -> MS),
OsmoBSC does mangle SAPI value. As a result, the MT transfer happens
on FACCH/F, while the MO transfer happens on SACCH/F. Such a mix!

#4 - 11/30/2018 12:56 PM - fixeria

- File mt_sms_bssap_rsl_lapdm.pcapng.gz added

#5 - 12/01/2018 03:26 PM - fixeria

- Project changed from OsmoMSC to OsmoBSC

- Subject changed from Use SACCH for MO/MT SMS during a voice call to Use SACCH for MO/MT SMS signalling during a voice call

- Category changed from SMS to A interface

- Priority changed from Low to High

- % Done changed from 0 to 90

It turns out the problem was in OsmoBSC, not in OsmoMSC. Please see:

https://gerrit.osmocom.org/#/c/osmo-bsc/+/12053

With this change applied, the whole signalling goes through SACCH, as expected.
Manually tested with two (physical, not virtual) phones and osmo-bts-trx.

#6 - 12/01/2018 08:31 PM - fixeria

- Status changed from Feedback to Resolved

- % Done changed from 90 to 100

Both patches have been merged, SMS goes over SACCH now.

#7 - 12/02/2018 09:54 AM - fixeria

- Tracker changed from Feature to Bug

- Status changed from Resolved to Stalled

- Assignee deleted (fixeria)

- Priority changed from High to Low

- % Done changed from 100 to 80

We still need a test case that should basically:

1) establish a dedicated TCH channel,
2) send a few messages on SAPI0 and make sure they appear on FACCH,
3) send a few messages on SAPI3 and make sure they appear on SACCH.

#8 - 05/08/2019 03:35 PM - laforge

- Assignee set to fixeria

#9 - 09/30/2020 08:07 PM - fixeria

- Status changed from Stalled to In Progress

#10 - 10/01/2020 04:49 PM - fixeria

- Status changed from In Progress to Feedback

- % Done changed from 80 to 100

I finally came up with a test case:

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20385 BSC_Tests: introduce TC_tch_dlci_link_id_sapi for OS#3716

this test case depends on the following misc/cosmetic changes:

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20382 BSC_Tests: s/f_verify_active_layer3/f_mo_l3_transceive/g
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20383 BSC_Tests: parametrize f_mo_l3_transceive()
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/20384 BSC_Tests: introduce f_mt_l3_transceive() sending BSSAP -> RSL

Of course, it reveals some problems in osmo-bsc: we cannot assign RSL Link ID to DLCI and vice versa, we need to convert:

https://gerrit.osmocom.org/c/osmo-bsc/+/20386 RSL/BSSAP: fix: properly convert between RSL Link ID and DLCI

and dissection problems in Wireshark:

https://gitlab.com/wireshark/wireshark/-/merge_requests/458/diffs?commit_id=9cef4e36294b3003c37b5369785e0105538573df

#11 - 10/03/2020 07:22 AM - fixeria

- Status changed from Feedback to Resolved

Files

mt_sms_bssap_rsl_lapdm.pcapng.gz	5.6 KB	11/30/2018	fixeria