OsmoMSC - Bug #3910

osmo-msc resends same sms with increased RP-MR after sending deliver_sm for the first one

04/09/2019 02:30 PM - pespin

Status:	Closed	**Start date:**	04/09/2019
Priority:	High	**Due date:**	
Assignee:	fixeria	**% Done:**	0%
Category:	SMS		
Target version:			
Resolution:			

Description

Reproducible 100% of times in osmo-gsm-tester test smpp/esme_ms_sms_storeforward.py:
https://git.osmocom.org/osmo-gsm-tester/tree/suites/smpp/esme_ms_sms_storeforward.py

Started failing as of March 10th, previous run (successful) was on March 7th:
https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_run-prod/1794/
https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_run-prod/1795/

An sms is sent from an ESME to an MS not yet registered to the network with Store&Forward mode and delivery receipt enabled. Then the MS is powered on and registers to network. Once registered, osmo-msc sends the SMS to hte MS, which gets it fine and sends the delivery report back. osmo-msc then sends the deliver_sm message to the ESME, which acks it. Until here verything's fine. But then immediately afterwards, osmo-msc sends again the sms to the MS (which receives it fine), this time with an increased RP-MR (0->1).

Changed commited to osmo-msc during that time:
8e2c6a31c1401b5a6980866ef35d47eb3d8d5ca3..f90496f577e78944ce8db1aa5b900477c1e479b0

Related issues:

Related to OsmoMSC - Bug #3912: memory leak in SMPP interface	**Resolved**	**04/09/2019**

History

#1 - 04/09/2019 04:06 PM - fixeria

- Status changed from New to In Progress

- Assignee changed from pespin to fixeria

Looking at the attached logs of OsmoMSC, I also noticed that the RAN connection, that was used for Location Updating, was not reused for MT SMS delivery. Instead, the Paging procedure was initiated (despite there was an active connection):

```
### Final part of Location Updating procedure
DMM gsm_04_08.c:1126 TMSI Reallocation Completed. Subscriber: IMSI-901700000009031:MSISDN-2014:TMSI-0x46C2D4D8
:TMSInew-0xA3B74039
DMM ran_conn.c:735 RAN_conn(IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039:GERAN-A-0:LU)[0x61200001f720]{RAN
_CONN_S_AUTH_CIPH}: Updated ID
DMM vlr_lu_fsm.c:741 RAN_conn(IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039:GERAN-A-0:LU)[0x61200001f720]{R
AN_CONN_S_AUTH_CIPH}: Received Event RAN_CONN_E_ACCEPTED

### Here we inform ESME that subscriber is available
DSMPP smpp_smsc.c:653 [esme-2013] Tx ALERT_NOTIFICATION (2014/3/1): Available

### Here we allocate a new transaction for MT SMS
DLSMS gsm_04_11.c:1064 Going to send a MT SMS
DCC transaction.c:118 (ti 00 sub IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039 callref 40000001) New transa
ction
DREF transaction.c:124 VLR subscr IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039 usage increases to: 4
DLSMS gsm0411_smc.c:95 SMC(0) instance created for network
DLSMS gsm0411_smr.c:91 SMR(0) instance created for network.
DLSMS gsm0411_smr.c:421 SMR(0) message SM-RL-DATA_REQ received in state IDLE
DLSMS gsm0411_smr.c:221 SMR(0) TX SMS RP-DATA
DLSMS gsm0411_smr.c:145 SMR(0) new RP state IDLE -> WAIT_FOR_RP_ACK
DLSMS gsm0411_smc.c:474 SMC(0) message MNSMS-EST-REQ received in state IDLE
DLSMS gsm0411_smc.c:141 SMC(0) new CP state IDLE -> MM_CONN_PENDING

### Despite we still have an active connection, OsmoMSC initiates the Paging procedure
```

```
DLSMS gsm_04_11.c:194 Initiating Paging procedure for IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039 due to
MMSMS_EST_REQ
DMM gsm_subscriber.c:163 Subscriber IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039 not paged yet, start pagi
ng.

### And now the existing connection becomes available for SMS delivery, but it's too late :/
DMM ran_conn.c:146 RAN_conn(IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039:GERAN-A-0:LU)[0x61200001f720]{RAN
_CONN_S_AUTH_CIPH}: state_chg to RAN_CONN_S_ACCEPTED
DREF osmo_msc.c:114 IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039: MSC conn use - dtap == 0 (0x0: )
DMM ran_conn.c:906 RAN_conn(IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039:GERAN-A-0:LU)[0x61200001f720]{RAN
_CONN_S_ACCEPTED}: Received Event RAN_CONN_E_UNUSED
DMM ran_conn.c:297 RAN_conn(IMSI-901700000009031:MSISDN-2014:TMSI-0xA3B74039:GERAN-A-0:LU)[0x61200001f720]{RAN
_CONN_S_ACCEPTED}: state_chg to RAN_CONN_S_RELEASING
```

After a short investigation, I found the reason why it was working before, and why doesn't work now:

```
struct ran_conn *connection_for_subscr(struct vlr_subscr *vsub)
{
        struct gsm_network *net = vsub->vlr->user_ctx;
        struct ran_conn *conn;

        llist_for_each_entry(conn, &net->ran_conns, entry) {
                if (conn->vsub != vsub)
                        continue;
                /* Found a conn, but is it in a usable state? Must not add transactions to a conn that is in r
elease,
                 * and must not start transactions for an unauthenticated subscriber. There will obviously be
only one
                 * conn for this vsub, so return NULL right away. */
                if (!ran_conn_is_accepted(conn))
                        return NULL;
                return conn;
        }

        return NULL;
}
```

At the moment we trigger the SMS queue, the existing RAN connection is still in state RAN_CONN_S_AUTH_CIPH!

#2 - 04/09/2019 04:32 PM - fixeria

> At the moment we trigger the SMS queue, the existing RAN connection is still in state RAN_CONN_S_AUTH_CIPH!

The SMS delivery is triggered by signal S_SUBSCR_ATTACHED. The only place where we dispatch this signal is in ran_conn.c:

https://git.osmocom.org/osmo-msc/tree/src/libmsc/ran_conn.c#n101

Function evaluate_acceptance_outcome() is getting called by the 'RAN_conn' FSM when one of the following events is received:

- RAN_CONN_E_ACCEPTED,
- RAN_CONN_E_MO_CLOSE,
- RAN_CONN_E_CN_CLOSE,

in the following way:

```
case RAN_CONN_E_ACCEPTED:
    evaluate_acceptance_outcome(fi, true);
    osmo_fsm_inst_state_chg(fi, RAN_CONN_S_ACCEPTED, RAN_CONN_TIMEOUT, 0);
    return;
```

So that's why the existing RAN connection remains in state RAN_CONN_S_AUTH_CIPH.
In any case, this problem is behind the scope of this issue. I'll look further.

#3 - 04/09/2019 04:55 PM - fixeria

Regarding the memory leak:

```
==5238==ERROR: LeakSanitizer: detected memory leaks
```

```
Direct leak of 3120 byte(s) in 3 object(s) allocated from:
    #0 0x7fc4e2df5d28 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.3+0xc1d28)
    #1 0x7fc4e12f8384 in smpp34_unpack ../def_frame/submit_sm.frame:19

Direct leak of 2080 byte(s) in 2 object(s) allocated from:
    #0 0x7fc4e2df5d28 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.3+0xc1d28)
    #1 0x7fc4e133d0a3 in build_tlv ../src/smpp34_params.c:105
    #2 0x61400000de9f  (<unknown module>)

Direct leak of 2080 byte(s) in 2 object(s) allocated from:
    #0 0x7fc4e2df5d28 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.3+0xc1d28)
    #1 0x7fc4e133d0a3 in build_tlv ../src/smpp34_params.c:105

Direct leak of 1040 byte(s) in 1 object(s) allocated from:
    #0 0x7fc4e2df5d28 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.3+0xc1d28)
    #1 0x7fc4e133d0a3 in build_tlv ../src/smpp34_params.c:105
    #2 0x61400000de9f  (<unknown module>)
    #3 0x5650fd41bb47 in alert_all_esme /home/osmocom-build/jenkins/workspace/osmo-gsm-tester_build-osmo-msc/o
smo-msc/src/libmsc/smpp_openbsc.c:303

SUMMARY: AddressSanitizer: 8320 byte(s) leaked in 8 allocation(s).
```

it looks more like a problem of libsmpp34.

#4 - 04/09/2019 09:55 PM - laforge

- Related to Bug #3912: memory leak in SMPP interface added

#5 - 04/12/2019 09:02 AM - fixeria

- Status changed from In Progress to Stalled

#6 - 05/26/2019 12:07 AM - fixeria

- Status changed from Stalled to Feedback

- Priority changed from Normal to High

pespin could you please check if the problem is still present in the new OsmoMSC? Thanks!

#7 - 05/27/2019 08:56 AM - pespin

Issue seems fixed since a few days ago, probably due to new OsmoMSC. Feel free to close the ticket fixeria .

#8 - 05/28/2019 01:35 AM - fixeria

- Status changed from Feedback to Closed

Files

test_run.tar.gz	149 KB	04/09/2019	pespin