OsmoMSC - Bug #3125

testcase for fixed "osmo-msc crashes while handling a call"

03/29/2018 03:34 PM - pespin

Status:	New	**Start date:**	03/29/2018
Priority:	Low	**Due date:**	
Assignee:	neels	**% Done:**	50%
Category:			
Target version:			
Resolution:			

Description

osmo-gsm-tester reported "osmo-msc process ended prematurely" rc=-6.

The process aborts due toan assertion in code.

This is the output from osmo-gsm-tester point of view:

```
16:54:15.129358 run              osmo-msc_10.42.42.6: DBG: attached: GET_REPLY 0 subscriber-list-a
ctive-v1 901700000015252,1758 901700000015253,1759   [trial-552↬voice:nanobts+band-900↬osmo-msc_10
.42.42.6]  [osmo_msc.py:129]
16:54:15.129741 tst                          /sierra_2: DBG: call_id_list: []  [trial-552↬voice:nano
bts+band-900↬/sierra_2]  [modem.py:680]
16:54:15.130064 tst                          /sierra_3: DBG: call_id_list: []  [trial-552↬voice:nano
bts+band-900↬/sierra_3]  [modem.py:680]
16:54:15.130308 tst                          /sierra_2: DBG: Dialing: 1759  [trial-552↬voice:nanobts
+band-900↬/sierra_2]  [modem.py:688]
16:54:15.264801 tst                          /sierra_2: DBG: Adding /sierra_2/voicecall01 to call li
st  [trial-552↬voice:nanobts+band-900↬/sierra_2]  [modem.py:692]
16:54:15.265427 tst                          /sierra_3: DBG: Waiting for incoming call from: 1758  [
trial-552↬voice:nanobts+band-900↬/sierra_3]  [modem.py:711]
16:54:15.268882 tst                          /sierra_2: DBG: 'org.ofono.VoiceCallManager'.CallAdded(
) -> /sierra_2/voicecall01="{'RemoteHeld': False, 'Emergency': False, 'State': 'dialing', 'LineIde
ntification': '1759', 'Name': '', 'RemoteMultiparty': False, 'Multiparty': False}"  [trial-552↬voi
ce:nanobts+band-900↬/sierra_2]  [modem.py:737]
16:54:15.269413 tst                          /sierra_2: DBG: Call already exists '/sierra_2/voiceca
l01'  [trial-552↬voice:nanobts+band-900↬/sierra_2]  [modem.py:741]
16:54:19.297451 tst                          /sierra_3: DBG: 'org.ofono.VoiceCallManager'.CallAdded(
) -> /sierra_3/voicecall01="{'RemoteHeld': False, 'Emergency': False, 'State': 'dialing', 'LineIde
ntification': '1758', 'Name': '', 'RemoteMultiparty': False, 'Multiparty': False}"  [trial-552↬voi
ce:nanobts+band-900↬/sierra_3]  [modem.py:737]
16:54:20.363164 tst                  mo_mt_call.py:43: dial success  [trial-552↬voice:nanobts+band-
900↬mo_mt_call.py:43]  [mo_mt_call.py:43]
16:54:20.460671 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
nanobts+band-900↬/sierra_2]  [modem.py:733]
16:54:20.470976 tst                          /sierra_3: DBG: call state: incoming  [trial-552↬voice:
nanobts+band-900↬/sierra_3]  [modem.py:733]
16:54:20.471509 tst                          /sierra_3: DBG: Answer call /sierra_3/voicecall01  [tri
al-552↬voice:nanobts+band-900↬/sierra_3]  [modem.py:716]
16:54:20.483867 tst                          /sierra_3: DBG: call state: incoming  [trial-552↬voice:
nanobts+band-900↬/sierra_3]  [modem.py:733]
16:54:20.813281 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
nanobts+band-900↬/sierra_2]  [modem.py:733]
16:54:21.827018 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
nanobts+band-900↬/sierra_2]  [modem.py:733]
16:54:22.839918 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
nanobts+band-900↬/sierra_2]  [modem.py:733]
16:54:23.853080 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
nanobts+band-900↬/sierra_2]  [modem.py:733]
16:54:24.867347 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
nanobts+band-900↬/sierra_2]  [modem.py:733]
16:54:25.880720 tst                          /sierra_2: DBG: call state: alerting  [trial-552↬voice:
```

```
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:26.894281 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:27.897140 tst                        /sierra_3: DBG: 'org.ofono.VoiceCallManager'.CallRemove
d() -> /sierra_3/voicecall01  [trial-552↦voice:nanobts+band-900↦/sierra_3]  [modem.py:744]
16:54:27.922407 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:28.938202 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:29.952036 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:30.965586 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:31.978774 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:32.993105 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:34.007221 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:35.020361 tst                        /sierra_2: DBG: call state: alerting  [trial-552↦voice:
nanobts+band-900↦/sierra_2]  [modem.py:733]
16:54:36.023000 run   osmo-msc_10.42.42.6(pid=22500): DBG: Cleanup  [trial-552↦voice:nanobts+band-
900↦osmo-msc_10.42.42.6↦osmo-msc_10.42.42.6(pid=22500)]  [process.py:151]
16:54:36.024190 run   osmo-msc_10.42.42.6(pid=22500): ERR: Terminated: ERROR {rc=-6}  [trial-552↦v
oice:nanobts+band-900↦osmo-msc_10.42.42.6↦osmo-msc_10.42.42.6(pid=22500)]  [process.py:134]
16:54:36.030331 run   osmo-msc_10.42.42.6(pid=22500): stderr:
 |         struct gsm_call                contains     32 bytes in   1 blocks (ref 0) 0x564c184406
10
|     sms                              contains      0 bytes in   1 blocks (ref 0) 0x564c18339130
|     msgb                             contains  17464 bytes in   8 blocks (ref 0) 0x564c183390c0
|         xua msg                          contains   2184 bytes in   1 blocks (ref 0) 0x564c18448bf
0
|         MGCP tx                          contains   4232 bytes in   1 blocks (ref 0) 0x564c1844501
0
|         xua msg                          contains   2184 bytes in   1 blocks (ref 0) 0x564c18446d2
0
|         SCCP User Primitive              contains   1160 bytes in   1 blocks (ref 0) 0x564c1844489
0
|         M3UA MTP-TRANSFER.ind            contains   2696 bytes in   1 blocks (ref 0) 0x564c1843a10
0
|         xUA Client Rx                    contains   2696 bytes in   1 blocks (ref 0) 0x564c184381b
0
|         xua_asp-xlm msgb                 contains   2312 bytes in   1 blocks (ref 0) 0x564c1843784
0
  [trial-552↦voice:nanobts+band-900↦osmo-msc_10.42.42.6↦osmo-msc_10.42.42.6(pid=22500)]  [process.
py:148]
```

And here the stderr of osmo-msc showing the assertion:

```
[0;m20180329165435231 [1;34mDLSS7[0;m <001d> osmo_ss7_hmrt.c:278 m3ua_hmdc_rx_from_l2(): dpc=2=0.0
.2 not local, message is for routing
[0;m20180329165435231 [1;34mDLSS7[0;m <001d> osmo_ss7_hmrt.c:227 Found route for dpc=2=0.0.2: pc=0
=0.0.0 mask=0x0=0.0.0 via AS as0 proto=m3ua
[0;m20180329165435231 [1;34mDLSS7[0;m <001d> osmo_ss7_hmrt.c:233 rt->dest.as proto is M3UA for dpc
=2=0.0.2
[0;m20180329165435231 [1;34mDLSS7[0;m <001d> m3ua.c:507 XUA_AS(as0)[0x564c1842c1b0]{AS_ACTIVE}: Re
ceived Event AS-TRANSFER.req
[0;m[1;33m20180329165435231 [1;34mDMM[0;m[1;33m <0002> osmo_msc.c:356 Subscr_Conn(1130850710)[0x56
4c18442ca0]{SUBSCR_CONN_S_COMMUNICATING}: Received Event SUBSCR_CONN_E_CN_CLOSE
[0;m[1;33m20180329165435231 [1;34mDMM[0;m[1;33m <0002> subscr_conn.c:217 Subscr_Conn(1130850710)[0
x564c18442ca0]{SUBSCR_CONN_S_COMMUNICATING}: state_chg to SUBSCR_CONN_S_RELEASED
[0;m[1;33m20180329165435232 [1;34mDMM[0;m[1;33m <0002> subscr_conn.c:245 Subscr_Conn(1130850710)[0
x564c18442ca0]{SUBSCR_CONN_S_RELEASED}: Terminating (cause = OSMO_FSM_TERM_REGULAR)
[0;m20180329165435232 [1;34mDVLR[0;m <000e> subscr_conn.c:245 Process_Access_Request_VLR(113085071
0)[0x564c18442dd0]{PR_ARQ_S_DONE}: Terminating (cause = OSMO_FSM_TERM_PARENT)
[0;m20180329165435232 [1;34mDVLR[0;m <000e> subscr_conn.c:245 Process_Access_Request_VLR(113085071
```

```
0)[0x564c18442dd0]{PR_ARQ_S_DONE}: Removing from parent Subscr_Conn(1130850710)[0x564c18442ca0]
[0;m20180329165435232 [1;34mDVLR[0;m <000e> subscr_conn.c:245 Process_Access_Request_VLR(1130850710)[0x564c18442dd0]{PR_ARQ_S_DONE}: Freeing instance
[0;m20180329165435232 [1;34mDVLR[0;m <000e> fsm.c:354 Process_Access_Request_VLR(1130850710)[0x564c18442dd0]{PR_ARQ_S_DONE}: Deallocated
[0;m[1;33m20180329165435232 [1;34mDMM[0;m[1;33m <0002> osmo_msc.c:344 msc_subscr_conn_close(vsub=MSISDN:1759, cause=2): no conn fsm, releasing directly without release event.
[0;m[1;32m20180329165435232 [1;34mDCC[0;m[1;32m <0001> gsm_04_08.c:1350 stopping pending timer T301
[0;m20180329165435232 [1;34mDMGCP[0;m <0007> msc_mgcp.c:1125 MGW(MGW_0)[0x564c18443340]{ST_MDCX_CN}: Received Event EV_TEARDOWN
[0;m20180329165435232 [1;31mDMGCP[0;m <0007> msc_mgcp.c:526 MGW(MGW_0)[0x564c18443340]{ST_MDCX_CN}: unexpected connection teardown -- graceful shutdown...
[0;m20180329165435232 [1;34mDMGCP[0;m <0007> msc_mgcp.c:190 MGW(MGW_0)[0x564c18443340]{ST_MDCX_CN}: state_chg to ST_CALL
[0;m20180329165435232 [1;34mDMGCP[0;m <0007> msc_mgcp.c:191 MGW(MGW_0)[0x564c18443340]{ST_CALL}: Received Event EV_TEARDOWN_ERROR
[0;m20180329165435232 [1;34mDMGCP[0;m <0007> msc_mgcp.c:736 MGW(MGW_0)[0x564c18443340]{ST_CALL}: DLCX: removing connection for the RAN and CN side on MGW endpoint:0x564c18443078...
[0;m20180329165435232 [1;32mDLMGCP[0;m <0021> mgcp_client.c:666 Queued 39 bytes for MGCP GW
[0;m20180329165435232 [1;34mDMGCP[0;m <0007> msc_mgcp.c:761 MGW(MGW_0)[0x564c18443340]{ST_CALL}: state_chg to ST_HALT
[0;m20180329165435232 [1;34mDMGCP[0;m <0007> msc_mgcp.c:1134 (subscriber:MSISDN:1759) call release initiated
[0;m[1;39m20180329165435232 [1;34mDMNCC[0;m[1;39m <0004> gsm_04_08.c:1362 transmit message MNCC_REL_IND
[0;m[1;32m20180329165435232 [1;34mDCC[0;m[1;32m <0001> gsm_04_08.c:1385 Sending 'MNCC_REL_IND' to MNCC.
[0;m[1;39m20180329165435232 [1;34mDMNCC[0;m[1;39m <0004> mncc_builtin.c:311 (call 1) Received message MNCC_REL_IND
[0;m[1;39m20180329165435232 [1;34mDMNCC[0;m[1;39m <0004> mncc_builtin.c:241 (call 1) Releasing remote with cause 47
[0;m[1;39m20180329165435232 [1;34mDMNCC[0;m[1;39m <0004> mncc_builtin.c:51 (call 1) Call removed.
[0;m[1;39m20180329165435232 [1;34mDMNCC[0;m[1;39m <0004> gsm_04_08.c:2975 receive message MNCC_REL_REQ
[0;m[1;32m20180329165435232 [1;34mDCC[0;m[1;32m <0001> gsm_04_08.c:3149 (ti 08 sub 1758) Received 'MNCC_REL_REQ' from MNCC in state 4 (CALL_DELIVERED)
[0;m[1;32m20180329165435232 [1;34mDCC[0;m[1;32m <0001> gsm_04_08.c:1626 starting timer T308 with 10 seconds
[0;m[1;32m20180329165435232 [1;34mDCC[0;m[1;32m <0001> gsm_04_08.c:1310 (ti 08 sub MSISDN:1758) new state CALL_DELIVERED -> RELEASE_REQ
[0;m20180329165435232 [1;34mDMSC[0;m <0006> msc_ifaces.c:60 msc_tx 6 bytes to MSISDN:1758 via RAN_GERAN_A
[0;m20180329165435232 [1;34mDBSSAP[0;m <0010> a_iface.c:155 (subscr MSISDN:1758, conn_id 4) Passing DTAP message from MSC to BSC
[0;m20180329165435232 [1;34mDBSSAP[0;m <0010> a_iface.c:169 (subscr MSISDN:1758, conn_id 4) N-DATA.req([])
[0;m20180329165435233 [1;34mDLSCCP[0;m <001e> sccp_scoc.c:1615 Received SCCP User Primitive N-DATA.request)
[0;m20180329165435233 [1;34mDLSCCP[0;m <001e> sccp_scoc.c:1657 SCCP-SCOC(4)[0x564c18440030]{ACTIVE}: Received Event N-DATA.req
[0;m20180329165435233 [1;34mDLSS7[0;m <001d> sccp_scrc.c:391 sccp_scrc_rx_scoc_conn_msg:  HDR=(CO: CODT,V=0,LEN=0),
    PART(T=Routing Context,L=4,D=00000000),
    PART(T=Destination Reference,L=4,D=00000005),
    PART(T=Data,L=9,D=010006832d080281af)
[0;m20180329165435233 [1;34mDLSS7[0;m <001d> osmo_ss7_hmrt.c:278 m3ua_hmdc_rx_from_l2(): dpc=2=0.0.2 not local, message is for routing
[0;m20180329165435233 [1;34mDLSS7[0;m <001d> osmo_ss7_hmrt.c:227 Found route for dpc=2=0.0.2: pc=0=0.0.0 mask=0x0=0.0.0 via AS as0 proto=m3ua
[0;m20180329165435233 [1;34mDLSS7[0;m <001d> osmo_ss7_hmrt.c:233 rt->dest.as proto is M3UA for dpc=2=0.0.2
[0;m20180329165435233 [1;34mDLSS7[0;m <001d> m3ua.c:507 XUA_AS(as0)[0x564c1842c1b0]{AS_ACTIVE}: Received Event AS-TRANSFER.req
[0;m[1;32m20180329165435233 [1;34mDCC[0;m[1;32m <0001> gsm_04_08.c:1310 (ti 00 sub MSISDN:1759) new state CALL_RECEIVED -> NULL
[0;mAssert failed conn->conn_fsm transaction.c:162
```

```
talloc report on 'vty' (total 136837 bytes in 7623 blocks)
    10.42.42.6                     contains     11 bytes in   1 blocks (ref 0) 0x564c1842b540
    save_cwd                       contains    140 bytes in   1 blocks (ref 0) 0x564c1834c830
    vty_command                    contains  81856 bytes in 4463 blocks (ref 0) 0x564c18339af0
    vty_vector                     contains  54830 bytes in 3157 blocks (ref 0) 0x564c18339a80
full talloc report on 'osmo_msc' (total  30129 bytes in  89 blocks)
    telnet_connection              contains      1 bytes in   1 blocks (ref 0) 0x564c1842b260
    10.42.42.6                     contains     11 bytes in   1 blocks (ref 0) 0x564c1842bb30
    struct osmo_ss7_instance       contains   4156 bytes in  39 blocks (ref 0) 0x564c1842b7c0
        struct osmo_sccp_instance      contains   2112 bytes in  13 blocks (ref 0) 0x564c184368f0
            struct sccp_connection         contains    927 bytes in   4 blocks (ref 0) 0x564c18441
d00
                struct osmo_fsm_inst           contains    223 bytes in   3 blocks (ref 0) 0x564c1
8442030
                    SCCP-SCOC(5)[0x564c18442030]   contains     29 bytes in   1 blocks (ref 0) 0x5
64c184421d0
                    5                              contains      2 bytes in   1 blocks (ref 0) 0x5
64c18442160
            struct sccp_connection         contains    927 bytes in   4 blocks (ref 0) 0x564c1843f
d00
                struct osmo_fsm_inst           contains    223 bytes in   3 blocks (ref 0) 0x564c1
8440030
                    SCCP-SCOC(4)[0x564c18440030]   contains     29 bytes in   1 blocks (ref 0) 0x5
64c184401d0
                    4                              contains      2 bytes in   1 blocks (ref 0) 0x5
64c18440160
            struct osmo_sccp_user          contains    154 bytes in   4 blocks (ref 0) 0x564c18436
9c0
                struct bsc_conn                contains     32 bytes in   1 blocks (ref 0) 0x564c1
8444f80
                struct bsc_conn                contains     32 bytes in   1 blocks (ref 0) 0x564c1
8441240
                OsmoMSC-A                      contains     10 bytes in   1 blocks (ref 0) 0x564c1
8435e30
        struct osmo_ss7_as             contains    560 bytes in   7 blocks (ref 0) 0x564c1842c060
            as0                            contains      4 bytes in   1 blocks (ref 0) 0x564c1842c
420
            struct osmo_fsm_inst           contains    328 bytes in   4 blocks (ref 0) 0x564c1842c
1b0
                struct xua_as_fsm_priv         contains    104 bytes in   1 blocks (ref 0) 0x564c1
842c350
                XUA_AS(as0)[0x564c1842c1b0]    contains     28 bytes in   1 blocks (ref 0) 0x564c1
842ba30
                as0                            contains      4 bytes in   1 blocks (ref 0) 0x564c1
842c2e0
            as0                            contains      4 bytes in   1 blocks (ref 0) 0x564c1842b
ac0
        struct osmo_ss7_asp            contains   1057 bytes in  14 blocks (ref 0) 0x564c1842bc50
            (r=10.42.42.5:2905<->l=10.42.42.1:56265) contains     41 bytes in   1 blocks (ref 0) 0
x564c18436a80
            struct osmo_fsm_inst           contains    331 bytes in   4 blocks (ref 0) 0x564c1842b
df0
                struct xua_asp_fsm_priv        contains    104 bytes in   1 blocks (ref 0) 0x564c1
842bf90
                XUA_ASP(asp0)[0x564c1842bdf0]  contains     30 bytes in   1 blocks (ref 0) 0x564c1
842bbc0
                asp0                           contains      5 bytes in   1 blocks (ref 0) 0x564c1
842bf20
            struct osmo_stream_cli         contains    235 bytes in   2 blocks (ref 0) 0x564c18436
7a0
                10.42.42.5                     contains     11 bytes in   1 blocks (ref 0) 0x564c1
8435ee0
            struct osmo_fsm_inst           contains    242 bytes in   4 blocks (ref 0) 0x564c18433
fc0
                struct lm_fsm_priv             contains      8 bytes in   1 blocks (ref 0) 0x564c1
8434180
                xua_default_lm(asp0)[0x564c18433fc0] contains     37 bytes in   1 blocks (ref 0) 0
```

```
x564c184340f0
                asp0                            contains      5 bytes in   1 blocks (ref 0) 0x564c1
8431c90
            10.42.42.5                          contains     11 bytes in   1 blocks (ref 0) 0x564c1842b
440
            asp0                                contains      5 bytes in   1 blocks (ref 0) 0x564c1842b
d80
        struct osmo_ss7_route_table    contains    131 bytes in   4 blocks (ref 0) 0x564c1842b6d0
            struct osmo_ss7_route               contains     68 bytes in   2 blocks (ref 0) 0x564c18433
f10
                as0                             contains      4 bytes in   1 blocks (ref 0) 0x564c1
8433b30
            system                              contains      7 bytes in   1 blocks (ref 0) 0x564c18381
430
    struct smsc                     contains    163 bytes in   2 blocks (ref 0) 0x564c18417d40
        10.42.42.6                      contains     11 bytes in   1 blocks (ref 0) 0x564c1842c5a0
    struct gsm_network              contains   7189 bytes in  32 blocks (ref 0) 0x564c18380b80
        struct osmo_fsm_inst            contains    243 bytes in   3 blocks (ref 0) 0x564c18442ca0
            Subscr_Conn(1130850710)[0x564c18442ca0] contains     40 bytes in   1 blocks (ref 0) 0x
564c18444600
            1130850710                          contains     11 bytes in   1 blocks (ref 0) 0x564c18443
820
        struct gsm_subscriber_connection contains    320 bytes in   1 blocks (ref 0) 0x564c1844438
0
        struct osmo_fsm_inst            contains    605 bytes in   7 blocks (ref 0) 0x564c18440750
            struct osmo_fsm_inst            contains    362 bytes in   4 blocks (ref 0) 0x564c18440
880
                struct proc_arq_priv            contains    104 bytes in   1 blocks (ref 0) 0x564c1
8440a50
                Process_Access_Request_VLR(1327048863)[0x564c18440880] contains     55 bytes in
1 blocks (ref 0) 0x564c184409b0
                1327048863                      contains     11 bytes in   1 blocks (ref 0) 0x564c1
8441ae0
            Subscr_Conn(1327048863)[0x564c18440750] contains     40 bytes in   1 blocks (ref 0) 0x
564c18441a50
            1327048863                          contains     11 bytes in   1 blocks (ref 0) 0x564c18441
9d0
        struct gsm_subscriber_connection contains    320 bytes in   1 blocks (ref 0) 0x564c184417d
0
        struct bsc_context              contains    472 bytes in   2 blocks (ref 0) 0x564c18432230
            struct a_reset_ctx              contains    288 bytes in   1 blocks (ref 0) 0x564c18433
1a0
        struct mgcp_client              contains    224 bytes in   2 blocks (ref 0) 0x564c18433da0
            struct mgcp_response_pending    contains     40 bytes in   1 blocks (ref 0) 0x564c18444
530
        struct gsm_sms_queue            contains    216 bytes in   1 blocks (ref 0) 0x564c18431890
        struct ctrl_handle              contains     80 bytes in   1 blocks (ref 0) 0x564c18429cf0
        10.42.42.2                      contains     11 bytes in   1 blocks (ref 0) 0x564c183814a0
        10.42.42.3                      contains     11 bytes in   1 blocks (ref 0) 0x564c1842b350
        osmo-gsm-tester-msc             contains     20 bytes in   1 blocks (ref 0) 0x564c1842b4c0
        osmo-gsm-tester-msc             contains     20 bytes in   1 blocks (ref 0) 0x564c1842b5c0
        struct vlr_instance             contains   2799 bytes in   8 blocks (ref 0) 0x564c18381520
            struct vlr_subscr               contains   1072 bytes in   1 blocks (ref 0) 0x564c1843e
5d0
            struct vlr_subscr               contains   1072 bytes in   1 blocks (ref 0) 0x564c1843c
fd0
            struct gsup_client              contains    487 bytes in   5 blocks (ref 0) 0x564c18430
fc0
                struct osmo_fd                  contains     48 bytes in   1 blocks (ref 0) 0x564c1
8431240
                struct ipa_client_conn          contains    187 bytes in   2 blocks (ref 0) 0x564c1
8431120
                    10.42.42.2                      contains     11 bytes in   1 blocks (ref 0) 0x5
64c184312e0
                MSC                             contains      4 bytes in   1 blocks (ref 0) 0x564c1
8430f50
        rate_ctr.c:228                  contains   1552 bytes in   1 blocks (ref 0) 0x564c18380d10
```

```
    transaction                     contains   1112 bytes in   2 blocks (ref 0) 0x564c18339210
        struct gsm_trans                contains   1112 bytes in   1 blocks (ref 0) 0x564c18440b20
    gsm_call                        contains     32 bytes in   2 blocks (ref 0) 0x564c183391a0
        struct gsm_call                 contains     32 bytes in   1 blocks (ref 0) 0x564c18440610
    sms                             contains      0 bytes in   1 blocks (ref 0) 0x564c18339130
    msgb                            contains  17464 bytes in   8 blocks (ref 0) 0x564c183390c0
        xua msg                         contains   2184 bytes in   1 blocks (ref 0) 0x564c18448bf0
        MGCP tx                         contains   4232 bytes in   1 blocks (ref 0) 0x564c18445010
        xua msg                         contains   2184 bytes in   1 blocks (ref 0) 0x564c18446d20
        SCCP User Primitive             contains   1160 bytes in   1 blocks (ref 0) 0x564c18444890
        M3UA MTP-TRANSFER.ind           contains   2696 bytes in   1 blocks (ref 0) 0x564c1843a100
        xUA Client Rx                   contains   2696 bytes in   1 blocks (ref 0) 0x564c184381b0
        xua_asp-xlm msgb                contains   2312 bytes in   1 blocks (ref 0) 0x564c18437840
```

I attach the osmo-gsm-teser run .tar.gz with all the output. You can also find a core file for osmo-msc in there. Interesting related directory is: run.2018-03-29_15-32-36/voice:nanobts+band-900/mo_mt_call.py/osmo-msc_10.42.42.6/

Related issues:

Related to OsmoMSC - Bug #3122: fix subscr_conn fsm: safely catch all compl-l...	**Resolved**	**03/28/2018**
Related to OsmoMSC - Bug #2880: OsmoMSC never releases MO call if MNCC doesn'...	**Resolved**	**01/26/2018**

History

#1 - 03/29/2018 03:34 PM - pespin

- Related to Bug #3122: fix subscr_conn fsm: safely catch all compl-l3 failures, properly handle all release situations added

#2 - 03/29/2018 03:34 PM - pespin

- Related to Bug #2880: OsmoMSC never releases MO call if MNCC doesn't respond to SETUP added

#3 - 03/29/2018 03:35 PM - pespin

- File trial-552-run.tgz added

Forgot to attach the run.tar.gz

#4 - 03/29/2018 03:39 PM - pespin

gdb bt with the core file:

```
Program terminated with signal SIGABRT, Aborted.
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:51
51      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) bt
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:51
#1  0x00007fd74557651a in __GI_abort () at abort.c:118
#2  0x0000564c167f701d in trans_free (trans=0x564c184438a0) at transaction.c:162
#3  0x0000564c167f7163 in trans_conn_closed (conn=conn@entry=0x564c18444380) at transaction.c:238
#4  0x0000564c167f7b02 in msc_subscr_conn_release_all (cause=<optimized out>, conn=0x564c18444380) at osmo_msc
.c:306
#5  msc_subscr_conn_close (conn=conn@entry=0x564c18444380, cause=<optimized out>) at osmo_msc.c:347
#6  0x0000564c167f8972 in subscr_conn_fsm_cleanup (fi=<optimized out>, cause=<optimized out>) at subscr_conn.c
:229
#7  0x00007fd74639eafa in ?? ()
#8  0x0000000000000003 in ?? ()
#9  0x00007fd7463ab6bc in ?? ()
#10 0x0000564c18444d20 in ?? ()
#11 0x0000000000000000 in ?? ()
```

#5 - 03/29/2018 03:44 PM - pespin

More complete bt after setting up lib dir correctly.

```
(gdb) bt
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:51
#1  0x00007fd74557651a in __GI_abort () at abort.c:118
#2  0x0000564c167f701d in trans_free (trans=0x564c184438a0) at transaction.c:162
```

```
#3  0x0000564c167f7163 in trans_conn_closed (conn=conn@entry=0x564c18444380) at transaction.c:238
#4  0x0000564c167f7b02 in msc_subscr_conn_release_all (cause=<optimized out>, conn=0x564c18444380) at osmo_msc
.c:306
#5  msc_subscr_conn_close (conn=conn@entry=0x564c18444380, cause=<optimized out>) at osmo_msc.c:347
#6  0x0000564c167f8972 in subscr_conn_fsm_cleanup (fi=<optimized out>, cause=<optimized out>) at subscr_conn.c
:229
#7  0x00007fd74639eafa in _osmo_fsm_inst_term (fi=0x564c18442ca0, cause=OSMO_FSM_TERM_REGULAR, data=0x0, file=
0x564c16811d84 "subscr_conn.c", line=245) at fsm.c:565
#8  0x00007fd74639e3ac in _osmo_fsm_inst_state_chg (fi=0x564c18442ca0, new_state=<optimized out>, timeout_secs
=0, T=0, file=<optimized out>, line=<optimized out>) at fsm.c:463
#9  0x00007fd74639e63d in _osmo_fsm_inst_dispatch (fi=0x564c18442ca0, event=event@entry=6, data=data@entry=0x7
ffea6b467fc, file=file@entry=0x564c16811153 "osmo_msc.c",
    line=line@entry=356) at fsm.c:517
#10 0x0000564c167f7a07 in msc_subscr_conn_close (conn=conn@entry=0x564c18444380, cause=<optimized out>, cause@
entry=1) at osmo_msc.c:356
#11 0x0000564c167f7b99 in msc_clear_request (conn=conn@entry=0x564c18444380, cause=cause@entry=1) at osmo_msc.
c:295
#12 0x0000564c167df0fc in bssmap_rx_clear_rqst (msg=<optimized out>, tp=0x7ffea6b46d40, conn=0x564c18444380) a
t a_iface_bssap.c:240
#13 rx_bssmap (scu=0x564c184369c0, a_conn_info=<optimized out>, msg=<optimized out>) at a_iface_bssap.c:590
#14 0x0000564c167dfb2a in a_sccp_rx_dt (scu=scu@entry=0x564c184369c0, a_conn_info=a_conn_info@entry=0x7ffea6b4
7d80, msg=<optimized out>) at a_iface_bssap.c:648
#15 0x0000564c167ddce4 in sccp_sap_up (oph=0x564c18444918, _scu=0x564c184369c0) at a_iface.c:566
#16 0x00007fd74639e63d in _osmo_fsm_inst_dispatch (fi=0x564c18442030, event=11, data=data@entry=0x564c184441f0
, file=file@entry=0x7fd745d1f43d "sccp_scoc.c",
    line=line@entry=1581) at fsm.c:517
#17 0x00007fd745d0fa5c in sccp_scoc_rx_from_scrc (inst=inst@entry=0x564c184368f0, xua=xua@entry=0x564c184441f0
) at sccp_scoc.c:1581
#18 0x00007fd745d0d6b0 in scrc_rx_mtp_xfer_ind_xua (inst=inst@entry=0x564c184368f0, xua=0x564c184441f0) at scc
p_scrc.c:449
#19 0x00007fd745d10635 in mtp_user_prim_cb (oph=0x564c1843a388, ctx=0x564c184368f0) at sccp_user.c:176
#20 0x00007fd745d08442 in m3ua_rx_xfer (xua=0x564c18439530, asp=0x564c1842bc50) at m3ua.c:586
#21 m3ua_rx_msg (asp=asp@entry=0x564c1842bc50, msg=msg@entry=0x564c184381b0) at m3ua.c:738
#22 0x00007fd745d1360b in xua_cli_read_cb (conn=<optimized out>) at osmo_ss7.c:1592
#23 0x00007fd7449782ab in osmo_stream_cli_read (cli=0x564c184367a0) at stream.c:192
#24 osmo_stream_cli_fd_cb (ofd=<optimized out>, what=1) at stream.c:276
#25 0x00007fd74639afde in osmo_fd_disp_fds (_eset=0x7ffea6b48190, _wset=0x7ffea6b48110, _rset=0x7ffea6b48090)
at select.c:216
#26 osmo_select_main (polling=<optimized out>) at select.c:256
#27 0x0000564c167dc454 in main (argc=3, argv=0x7ffea6b48358) at msc_main.c:694

(gdb) frame
#2  0x0000564c167f701d in trans_free (trans=0x564c184438a0) at transaction.c:162
162     in transaction.c
(gdb) print *trans
$1 = {entry = {next = 0x0, prev = 0x7fd742d96e68}, net = 0x7fd742d96e78, protocol = 232 '\350', transaction_id
 = 110 'n', vsub = 0x7fd742d96ef8, conn = 0x7fd742d96f08,
  callref = 1121545864, tch_recv = 32727, paging_request = 0x7fd742d96e98, bearer_cap = {transfer = 1121545720
, mode = 32727, coding = 1121545736, radio = 32727,
    speech_ctm = 1121545704, speech_ver = {32727, 0, 0, 0, 0, 0, 0, 0}, data = {rate_adaption = GSM48_BCAP_RA_
NONE, sig_access = 0, async = 0, nr_stop_bits = 1121545928,
      nr_data_bits = 32727, user_rate = 0, parity = GSM48_BCAP_PAR_ODD, interm_rate = 0, transp = GSM48_BCAP_T
R_TRANSP, modem_type = 1121545944}}, assignment_done = 215,
  tch_rtp_create = 127, {cc = {state = 0, Tcurrent = 0, T308_second = 1121545752, timer = {node = {rb_parent_c
olor = 140562516241976, rb_right = 0x7fd742d96e28,
          rb_left = 0x7fd742d96e48}, list = {next = 0x0, prev = 0x0}, timeout = {tv_sec = 0, tv_usec = 0}, act
ive = 0, cb = 0x7fd742d96f48, data = 0x7fd742d96f38}, msg = {
        msg_type = 1121546024, callref = 32727, fields = 1121546008, bearer_cap = {transfer = 32727, mode = 0,
 coding = 0, radio = 0, speech_ctm = 0, speech_ver = {
            1121546088, 32727, 0, 0, 0, 0, 0, 0}, data = {rate_adaption = GSM48_BCAP_RA_NONE, sig_access = 0,
async = 0, nr_stop_bits = 0, nr_data_bits = 0, user_rate = 0,
          parity = GSM48_BCAP_PAR_ODD, interm_rate = 0, transp = GSM48_BCAP_TR_TRANSP, modem_type = GSM48_BC
AP_MT_NONE}}, called = {type = 1121546072, plan = 32727,
          present = 0, screen = 0, number = '\000' <repeats 32 times>}, calling = {type = 0, plan = 0, present
 = 0, screen = 0, number = '\000' <repeats 32 times>},
        redirecting = {type = 0, plan = 0, present = 0, screen = 0, number = '\000' <repeats 32 times>}, conne
cted = {type = 0, plan = 0, present = 0, screen = 0,
          number = '\000' <repeats 32 times>}, cause = {location = 1121545816, coding = 32727, rec = 111935904
0, rec_val = 32727, value = 1119369872, diag_len = 32727,
          diag = "\a\000\037\000\000\000\000\000\300\021D\030LV\000\000\003\000\000\000\000\000\000\000\260<D\
030LV\000"}, progress = {coding = 0, location = 0, descr = 0},
        useruser = {proto = 0,
          info = " MD\030LV\000\000\017\000\000\000\203\000\000\000\017\000\000\000\n\000\000\000\000\022\270B
\327\177\000\000\200\022\270B\327\177\000\000\064\024\270B\327\177\000\000\001\000\000\000\016@\001", '\000' <
repeats 25 times>, "\216/\270B\327\177\000\000`*C\030LV\000\000\000\020\270B\327\177\000\000\060t\331B\327\177
```

```
\000\000\000p\271B\327\177\000\000(\244\006G\327\177\000", <incomplete sequence \370>}, facility = {len = 2209
2,
          info = '\000' <repeats 16 times>, "\004\000\000\000\000\000\000\000\230;D\030LV\000\000\370:D\030LV\
000\000\000\000\000\000\000\000\000\000\005\b\000\000\000\000\000\000\220\r\b", '\000' <repeats 21 times>, "\2
40\021D\030LV", '\000' <repeats 14 times>, "\001", '\000' <repeats 26 times>}, cccap = {dtmf = 0, pcp = 0}, ss
version = {len = 0,
          info = "\000\000\000\000\300\030\270B\327\177\000\000\004\000\000\000\000\000\000\000\000\260\004G\3
27\177\000\000(\rUE\327\177", '\000' <repeats 66 times>, "\270]!\000\000\000\000\000H\002\000\000\000\000\000\
000\037\000\000\000\000\000\000\000\000\000\000"}, clir = {sup = 0, inv = 407125200}, signal = 22092, keypad =
 0, more = 0,
        notify = 1, emergency = 0, imsi = "libgcc_s.so.1\000\000", lchan_type = 0 '\000', lchan_mode = 0 '\000
'}}, sms = {smc_inst = {id = 0, network = 1121545752,
        mn_recv = 0x7fd742d96e38, mm_send = 0x7fd742d96e28, cp_state = 1121545800, cp_timer = {node = {rb_pare
nt_color = 0, rb_right = 0x0, rb_left = 0x0}, list = {
            next = 0x0, prev = 0x0}, timeout = {tv_sec = 140562516242248, tv_usec = 140562516242232}, active =
 0, cb = 0x7fd742d96f18, data = 0x0}, cp_msg = 0x0,
        cp_rel = 1121546088, cp_retx = 32727, cp_max_retr = 0, cp_tc1 = 0}, smr_inst = {id = 0, network = 0, r
l_recv = 0x0, mn_send = 0x0, rp_state = GSM411_RPS_IDLE,
        rp_timer = {node = {rb_parent_color = 0, rb_right = 0x0, rb_left = 0x7fd742d96f58}, list = {next = 0x0
, prev = 0x0}, timeout = {tv_sec = 0, tv_usec = 0}, active = 0,
          cb = 0x0, data = 0x0}}, sms = 0x0}}, bridge = {peer = 0x0, state = BRIDGE_STATE_NONE}}
```

#6 - 03/29/2018 03:57 PM - neels

```
Assert failed conn->conn_fsm transaction.c:162
```

This assert is new, and it shows that the transaction is cleared only after the FSM is gone, which is unexpected.
This, too, should be solved by #3122.

#7 - 04/03/2018 09:14 AM - fixeria

Faced this issue (with a bit different bt) during external USSD implementation.
The problem was that I forgot to manually close / clean-up a transaction, so:

```
<0000> gsm_04_08.c:3463 Dispatching 04.08 message GSM48_MT_MM_CM_SERV_REQ (0x5:0x24)
<0002> gsm_04_08.c:711 <- CM SERVICE REQUEST serv_type=0x08 MI(TMSI)=716522206
<0002> fsm.c:272 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_INIT}: Allocated
<0002> subscr_conn.c:353 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_INIT}: Received Event SUBSCR_CONN_E_ST
ART
<0002> subscr_conn.c:67 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_INIT}: state_chg to SUBSCR_CONN_S_NEW
<000e> fsm.c:272 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: Allocated
<000e> fsm.c:302 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: is child of Subscr_Conn(71652
2206)[0x782250]
<000e> vlr_access_req_fsm.c:704 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: rev=GSM net=GE
RAN (no Auth)
<000e> vlr_access_req_fsm.c:730 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: Received Event
 PR_ARQ_E_START
<000e> vlr_access_req_fsm.c:337 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: proc_arq_vlr_f
n_post_imsi()
<000e> vlr_access_req_fsm.c:289 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: _proc_arq_vlr_
node2()
<000e> vlr_access_req_fsm.c:255 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: _proc_arq_vlr_
node2_post_ciph()
<000e> vlr_access_req_fsm.c:227 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: _proc_arq_vlr_
node2_post_vlr()
<000e> vlr_access_req_fsm.c:212 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: _proc_arq_vlr_
post_pres()
<000e> vlr_access_req_fsm.c:196 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: _proc_arq_vlr_
post_trace()
<000e> vlr_access_req_fsm.c:174 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: _proc_arq_vlr_
post_imei()
<000e> vlr_access_req_fsm.c:187 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: proc_arq_fsm_d
one(VLR_PR_ARQ_RES_PASSED)
<000e> vlr_access_req_fsm.c:109 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_INIT}: state_chg to P
R_ARQ_S_DONE
<000e> vlr_access_req_fsm.c:119 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_DONE}: Process Access
 Request result: VLR_PR_ARQ_RES_PASSED
<0002> msc_ifaces.c:103 -> CM SERVICE ACCEPT MSISDN:01393
<0002> vlr_access_req_fsm.c:158 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_NEW}: Received Event SUBSCR_CON
N_E_ACCEPTED
<0002> subscr_conn.c:78 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_NEW}: SUBSCR_CONN_FROM_CM_SERVICE_REQ
<0002> subscr_conn.c:85 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_NEW}: state_chg to SUBSCR_CONN_S_ACCEPT
```

```
ED
<0002> subscr_conn.c:130 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: received_cm_service_request
 = true
<0002> subscr_conn.c:133 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: Received Event SUBSCR_CONN_
E_RELEASE_WHEN_UNUSED
<0002> subscr_conn.c:148 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: subscr_conn_fsm_release_whe
n_unused: still awaiting first req
uest after a CM Service Request
<0002> osmo_msc.c:100 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: Received Event SUBSCR_CONN_E_R
ELEASE_WHEN_UNUSED
<0002> subscr_conn.c:148 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: subscr_conn_fsm_release_whe
n_unused: still awaiting first req
uest after a CM Service Request
```

Got the first message after CM Service Request, USSD Request:

```
<0000> gsm_04_08.c:3463 Dispatching 04.08 message NCSS:0x3b (0xb:0x3b)
<0002> ussd.c:274 Received SS/USSD data (trans_id=8)
```

Requested transaction doesn't exist, so we allocate (in my new code) a new one:

```
<0002> ussd.c:291  -> (new transaction)
<0001> transaction.c:96 (ti 08 sub MSISDN:01393 callref 0) New transaction
```

Calling cm_service_request_concludes() here:

```
<0002> gsm_04_08.c:3347 MSISDN:01393: rx msg NCSS:0x3b: received_cm_service_request changes to false
```

USSD handler responds:

```
<0002> ussd.c:177 USSD: Own number requested
<0002> ussd.c:142 MSISDN:01393: MSISDN = 01393
```

Calling msc_subscr_conn_communicating() in order to keep the connection:

```
<0002> subscr_conn.c:378 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: Received Event SUBSCR_CONN_
E_COMMUNICATING
<0002> subscr_conn.c:184 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_ACCEPTED}: state_chg to SUBSCR_CONN_S_
COMMUNICATING
<0002> osmo_msc.c:100 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_COMMUNICATING}: Received Event SUBSCR_CON
N_E_RELEASE_WHEN_UNUSED
<0002> subscr_conn.c:166 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_COMMUNICATING}: subscr_conn_fsm_releas
e_when_unused: connection still has active transaction: NCSS
```

Some pause happens here, because I didn't call trans_free() manually!!!
Then, after some time, BSC requests MSC to clear the connection, see bt:

```
#10 0x000000000042c1a8 in subscr_conn_fsm_communicating (fi=0x782250, event=SUBSCR_CONN_E_CN_CLOSE, data=0x7ff
fffffc3f4) at subscr_conn.c:217
#11 0x00007ffff733676d in _osmo_fsm_inst_dispatch (fi=0x782250, event=SUBSCR_CONN_E_CN_CLOSE, data=0x7fffffffc
3f4, file=0x44dce3 "osmo_msc.c", line=0x164)
    at fsm.c:517
#12 0x000000000042ac14 in msc_subscr_conn_close (conn=0x7832d0, cause=0x0) at osmo_msc.c:356
#13 0x000000000042a9b7 in msc_clear_request (conn=0x7832d0, cause=0x0) at osmo_msc.c:295
#14 0x0000000000409c35 in bssmap_rx_clear_rqst (conn=0x7832d0, msg=0x782ae0, tp=0x7fffffffc4e0) at a_iface_bss
ap.c:240
#15 0x000000000040b13e in rx_bssmap (scu=0x767340, a_conn_info=0x7fffffffd570, msg=0x782ae0) at a_iface_bssap.
c:590
```

Here we run the following code:

```
static void subscr_conn_fsm_communicating(struct osmo_fsm_inst *fi,
uint32_t event, void *data)
{
  // ...

  /* Whatever unexpected happens in the accepted state, it means release.
   * Even if an unexpected event is passed, the safest thing to do is
   * discard the conn. We don't expect another SUBSCR_CONN_E_ACCEPTED. */
  osmo_fsm_inst_state_chg(fi, SUBSCR_CONN_S_RELEASED, 0, 0);
}
```

The corresponding logging part:

```
<0002> osmo_msc.c:356 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_COMMUNICATING}: Received Event SUBSCR_CON
```

```
N_E_CN_CLOSE
<0002> subscr_conn.c:217 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_COMMUNICATING}: state_chg to SUBSCR_CO
NN_S_RELEASED
<0002> subscr_conn.c:245 Subscr_Conn(716522206)[0x782250]{SUBSCR_CONN_S_RELEASED}: Terminating (cause = OSMO_F
SM_TERM_REGULAR)
<000e> subscr_conn.c:245 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_DONE}: Terminating (cause =
OSMO_FSM_TERM_PARENT)
<000e> subscr_conn.c:245 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_DONE}: Removing from parent
Subscr_Conn(716522206)[0x782250]
<000e> subscr_conn.c:245 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_DONE}: Freeing instance
<000e> fsm.c:354 Process_Access_Request_VLR(716522206)[0x782380]{PR_ARQ_S_DONE}: Deallocated
```

Changing state to SUBSCR_CONN_S_RELEASED causes a call to subscr_conn_fsm_release():

```
#7  0x00007ffff7336c6b in _osmo_fsm_inst_term (fi=0x782250, cause=OSMO_FSM_TERM_REGULAR, data=0x0, file=0x44f4
50 "subscr_conn.c", line=0xf5)
    at fsm.c:565
#8  0x000000000042c2b7 in subscr_conn_fsm_release (fi=0x782250, prev_state=0x3) at subscr_conn.c:245
#9  0x00007ffff73364bf in _osmo_fsm_inst_state_chg (fi=0x782250, new_state=SUBSCR_CONN_S_RELEASED, timeout_sec
s=0x0, T=0x0, file=<optimized out>,
    line=<optimized out>) at fsm.c:463
```

This triggers the termination call-back:

```
static void subscr_conn_fsm_cleanup(struct osmo_fsm_inst *fi,
    enum osmo_fsm_term_cause cause)
{
    struct gsm_subscriber_connection *conn = fi->priv;
    fi->priv = NULL;

    if (!conn)
        return;

    /* PAY YOR ATTENTION!!! */
    conn->conn_fsm = NULL;

    /* Here we CALLING IT AGAIN, it was already called! */
    msc_subscr_conn_close(conn, cause);
    msc_subscr_conn_put(conn, MSC_CONN_USE_FSM);
}
```

And here the magic happens. This then executes the following code:

```
void msc_subscr_conn_close(struct gsm_subscriber_connection *conn,
    uint32_t cause)
{
    // ...
    if (!conn->conn_fsm) {
        DEBUGP(DMM, "msc_subscr_conn_close(vsub=%s, cause=%u): no conn fsm,"
               " releasing directly without release event.\n",
               vlr_subscr_name(conn->vsub), cause);
        /* In case of an IMSI Detach, we don't have conn_fsm. Release
         * anyway to ensure a timely Iu Release / BSSMAP Clear. */
        msc_subscr_conn_release_all(conn, cause);
        return;
    }
    // ...
}
```

Then:

```
static void msc_subscr_conn_release_all(struct gsm_subscriber_connection *conn,
    uint32_t cause)
{
    // ...

    /* If we're closing in a middle of a trans, we need to clean up */
    trans_conn_closed(conn);

    // ...
}
```

But at the moment we have 'conn->conn_fsm = NULL'! So, the pending bt:

```
#0  0x00007ffff6729c37 in __GI_raise (sig=sig@entry=0x6) at ../nptl/sysdeps/unix/sysv/linux/raise.c:56
```

```
#1  0x00007ffff672d028 in __GI_abort () at abort.c:89
#2  0x0000000000429b92 in trans_free (trans=0x782620) at transaction.c:165
#3  0x0000000000429dd5 in trans_conn_closed (conn=0x7832d0) at transaction.c:241
#4  0x000000000042a9f2 in msc_subscr_conn_release_all (conn=0x7832d0, cause=0x2) at osmo_msc.c:306
#5  0x000000000042ab7f in msc_subscr_conn_close (conn=0x7832d0, cause=0x2) at osmo_msc.c:347
#6  0x000000000042c211 in subscr_conn_fsm_cleanup (fi=0x782250, cause=OSMO_FSM_TERM_REGULAR) at subscr_conn.c:
229
```

#8 - 04/03/2018 11:32 AM - neels

fixeria, I appreciate you looking into this; the short answer is the osmo-msc subscr conn FSM needs/needed a review to address this and various similar issues.
Over the weekend I've thoroughly refactored allocation and release, and now definitely all transactions must be cleaned up before deallocation.
May not be a magic wand to fix everything, but this situation should be handled properly after the patch.

The patch is not yet on gerrit, still needs some testing (and then code review).
Hope to conclude it ASAP, since more and more people seem to lose time on similar issues recently.

#9 - 04/07/2018 10:41 PM - neels

Can't get this crash reproduced, can you help me understand what else this test should be doing to trigger the crash?

```
private function f_tc_mo_cc_mo_cancel(charstring id, BSC_ConnHdlrPars pars) runs on BSC_ConnHdlr {
        f_init_handler(pars);
        var CallParameters cpars := valueof(t_CallParams('12345'H, 0));
        var MNCC_PDU mncc;
        var MgcpCommand mgcp_cmd;

        f_perform_lu();

        f_establish_fully(valueof(ts_MI_IMSI_LV(g_pars.imsi)));
        f_create_mncc_expect(hex2str(cpars.called_party));
        f_create_mgcp_expect(ExpectCriteria:{omit,omit,omit});

        BSSAP.send(ts_PDU_DTAP_MO(ts_ML3_MO_CC_SETUP(cpars.transaction_id, cpars.called_party)));
        MNCC.receive(tr_MNCC_SETUP_ind(?, tr_MNCC_number(hex2str(cpars.called_party)))) -> value mncc;
        cpars.mncc_callref := mncc.u.signal.callref;
        MNCC.send(ts_MNCC_CALL_PROC_req(cpars.mncc_callref, cpars.mncc_bearer_cap));
        BSSAP.receive(tr_PDU_DTAP_MT(tr_ML3_MT_CC_CALL_PROC(cpars.transaction_id)));

        MNCC.send(ts_MNCC_ALERT_req(cpars.mncc_callref));
        BSSAP.receive(tr_PDU_DTAP_MT(tr_ML3_MT_CC_ALERTING(cpars.transaction_id)));

        f_sleep(1.0);
        MNCC.send(ts_MNCC_REL_req(cpars.mncc_callref, valueof(ts_MNCC_cause(47))));

        BSSAP.receive(tr_PDU_DTAP_MT(tr_ML3_MT_CC_RELEASE(cpars.transaction_id)));
        BSSAP.send(ts_PDU_DTAP_MO(ts_ML3_MO_CC_REL_COMPL(cpars.transaction_id)));

        alt{
        [] MGCP.receive(tr_DLCX(?)) -> value mgcp_cmd {
                MGCP.send(ts_DLCX_ACK2(mgcp_cmd.line.trans_id));
                f_create_mgcp_delete_ep(cpars.mgcp_ep);
                repeat;
                }
        [] as_clear_cmd_compl_disc();
        }
}
```

#10 - 04/09/2018 08:05 AM - pespin

I didn't look into the details yet, but I can still see osmo-msc currently crashing in osmo-gsm-tester half of the times the mo_mt_call.py test is running. However, most of the times (or always) the issue appears when using a nanoBTS. I'd advise you to go check any of the run.tar.gz from osmo-gsm-tester where the issue is reproduced, then have a look at the pcap file for the osmo-msc for the test causing the crash to see the order of packets required.

#11 - 04/09/2018 08:08 AM - pespin

You should actually look at the pcap file under osmo-stp, that one will contain the BSSAP trace.

#12 - 04/11/2018 12:20 PM - neels

- % Done changed from 0 to 50

the bug should be fixed by https://gerrit.osmocom.org/#/c/7702/ and be long gone with https://gerrit.osmocom.org/7705
however, I still want a test case to trigger it...

#13 - 06/23/2018 07:26 PM - laforge

- Subject changed from osmo-msc crashes while handling a call to testcase for fixed "osmo-msc crashes while handling a call"

- Priority changed from Normal to Low

Files

trial-552-run.tgz	7.76 MB	03/29/2018	pespin