OsmoPCU - Bug #1756

regression: "tbf: Add state WAIT_ASSIGN"

06/16/2016 12:56 PM - neels

Status:	New	**Start date:**	06/16/2016
Priority:	Low	**Due date:**	
Assignee:	sysmocom	**% Done:**	0%
Category:			
Target version:			
Spec Reference:			

Description

Current osmo-pcu master is not working for GPRS on neither litecell15 nor sysmoBTS.

I've traced it to commit "tbf: Add state WAIT_ASSIGN" f1a7b8fc6651f92a8b7f3f27b7ca05d07f4e44e0

See https://gerrit.osmocom.org/218

un-duplicating issues:

- Please refer here for why the WAIT_ASSIGN state in f1a7b8fc66 breaks GPRS completely.
- Discuss reliability after revert of f1a7b8fc66 in #1742.

Related issues:

Related to OsmoPCU - Bug #1742: Regression in egprs patch series	**New**	**06/02/2016**

History

#1 - 06/16/2016 01:01 PM - neels

The error observed is the following log output (was viewing both osmo-bts-sysmo and osmo-pcu logs interleaved):

```
<0007> l1sap.c:920 RACH for packet access
<0009> pcu_sock.c:353 Sending RACH indication: qta=0, ra=123, fn=60633
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=123, fn=60633
<0009> pcu_sock.c:463 Data request received: sapi=AGCH arfcn=0 block=0 data=2d 06 3f 10 0e e3 64 7b 6d a1 00 0
0 c8 00 30 0b 2b 2b 2b 2b 2b 2b 2b
<0002> tbf.cpp:874 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) T3169 timeout during transsmission
<0002> tbf.cpp:893 - Assignment was on CCCH
<0002> tbf.cpp:899 - No uplink data received yet
[repeat]
```

An attached phone does not show the GPRS icon, so no data service is even offered.

#2 - 06/16/2016 01:04 PM - neels

On gerrit, hwelte said:
"Let's review the radisys osmo-pcu repository and coordinate with them on what they did to overcome this issue. Maybe they have a proper fix rather than reverting the WAIT_ASSIGN state altogether."

I have just checked out the radisys osmo-pcu repository and tested with it.

In it I find one branch being noteworthy: radisys/egprs_integration
Testing with that branch yields the same result: GPRS is not working.
Maybe EGPRS is not affected by this issue?

When I rebase the radisys branch onto the revert of the WAIT_ASSIGN commit, GPRS is working again.

#3 - 06/16/2016 01:05 PM - neels

TS config during testing:

```
 trx 0
  rf_locked 0
  arfcn 868
  nominal power 23
  max_power_red 10
  rsl e1 tei 0
  timeslot 0
   phys_chan_config CCCH+SDCCH4
   hopping enabled 0
  timeslot 1
   phys_chan_config SDCCH8
   hopping enabled 0
  timeslot 2
   phys_chan_config TCH/F
   hopping enabled 0
  timeslot 3
   phys_chan_config TCH/F
   hopping enabled 0
  timeslot 4
   phys_chan_config TCH/F
   hopping enabled 0
  timeslot 5
   phys_chan_config PDCH
   hopping enabled 0
  timeslot 6
   phys_chan_config PDCH
   hopping enabled 0
  timeslot 7
   phys_chan_config PDCH
   hopping enabled 0
```

#4 - 06/16/2016 01:06 PM - neels

So we still need to understand

- why the WAIT_ASSIGN state is never left
- what is achieved by this state, i.e. what breaks if we remove this state

#5 - 06/16/2016 01:44 PM - neels

commit log says

```
    tbf: Add state WAIT_ASSIGN

    Currently the state on the TBF is set to ASSIGN when it is created
    and in general changed to FLOW when it is acknowledged by the MS. The
    moment when the assignment is really transmitted to the MS (which can
    take some time) is not reflected by the state. A TBF can considered
    to be valid, when the assignment is received by the MS.

    Add the state WAIT_ASSIGN that is entered when the assigment has been
    send to the MS (more precisely: when the corresponding RTS has been
    processed or the message has been sent to the BTS).

    The TBF, its PDCH(s), and its TFI can be assumed to be valid, when the
    TBF has a state of WAIT_ASSIGN or higher (assuming that the TBF
    starting time has not been set, which is currently only done for SBA,
    which are not associated with a TBF).

    Note that due to queuing in the BTS there can still be a invalid
    time period for immediate assignments, when the request is lingering
    in the AGCH or PCH queues.
```

So the WAIT_ASSIGN state is entered when the PACCH assignment is sent out.
It seems we should receive an ack and then advance to the FLOW state.

The only state advances to FLOW are in bts.cpp:

- BTS::rcv_rach()
- gprs_rlcmac_pdch::rcv_control_ack()

It seems rcv_rach() is the place where the state should advance to FLOW.
A RLCMAC debug log looks like this:

```
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=122, fn=1697
<0002> bts.cpp:485 MS requests UL TBF on RACH, so we provide one:
<0002> tbf.cpp:672 ********** TBF starts here **********
<0002> tbf.cpp:674 Allocating UL TBF: MS_CLASS=0/0
<0002> gprs_ms.cpp:114 Creating MS object, TLLI = 0x00000000
<0002> bts.cpp:413 Searching for first unallocated TFI: TRX=0
<0002> bts.cpp:423  Found TFI=0.
<0002> gprs_rlcmac_ts_alloc.cpp:525 Slot Allocation (Algorithm B) for unknown class (assuming 12)
<0002> gprs_rlcmac_ts_alloc.cpp:560 - Rx=4 Tx=4 Sum Rx+Tx=5  Tta=2 Ttb=1  Tra=2 Trb=1 Type=1
<0002> gprs_rlcmac_ts_alloc.cpp:157 - Skipping TS 0, because not enabled
<0002> gprs_rlcmac_ts_alloc.cpp:157 - Skipping TS 1, because not enabled
<0002> gprs_rlcmac_ts_alloc.cpp:157 - Skipping TS 2, because not enabled
<0002> gprs_rlcmac_ts_alloc.cpp:157 - Skipping TS 3, because not enabled
<0002> gprs_rlcmac_ts_alloc.cpp:157 - Skipping TS 4, because not enabled
<0002> gprs_rlcmac_ts_alloc.cpp:579 - Possible DL/UL slots: (TS=0)".....CCC"(TS=7)
<0002> gprs_rlcmac_ts_alloc.cpp:278 - Skipping TS 7, because num TBFs 0 >= 0
<0002> gprs_rlcmac_ts_alloc.cpp:940 - Selected UL slots: (TS=0)"......Uu"(TS=7), single
<0002> gprs_rlcmac_ts_alloc.cpp:966 Using single slot at TS 6 for UL
<0002> gprs_rlcmac_ts_alloc.cpp:990 - Reserved DL/UL slots: (TS=0)"......C."(TS=7)
<0002> gprs_rlcmac_ts_alloc.cpp:1017 - Assigning UL TS 6
<0002> bts.cpp:1489 PDCH(TS 6, TRX 0): Attaching TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=NULL), 1 TBFs, USFs =
01, TFIs = 00000001.
<0002> tbf.cpp:385 - Setting Control TS 6
```

```
<0002> gprs_ms.cpp:267 Attaching TBF to MS object, TLLI = 0x00000000, TBF = TBF(TFI=0 TLLI=0x00000000 DIR=UL S
TATE=NULL)
<0002> tbf.cpp:625 Allocated TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=NULL): trx = 0, ul_slots = 40, dl_slots =
00
<0002> bts.cpp:291 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=NULL) changes state from NULL to FLOW
<0002> tbf.cpp:409 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=FLOW) starting timer 3169.
<0002> bts.cpp:529 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=FLOW) [UPLINK] START
<0002> bts.cpp:533 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=FLOW) RX: [PCU <- BTS] RACH qbit-ta=0 ra=0x7a, Fn=16
97 (1,14,7)
<0002> bts.cpp:535 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=FLOW) TX: START Immediate Assignment Uplink (AGCH)
<0002> bts.cpp:548  - TRX=0 (868) TS=6 TA=0 TSC=7 TFI=0 USF=0
<0002> bts.cpp:291 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=FLOW) changes state from FLOW to WAIT ASSIGN
<0002> tbf.cpp:819 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) timer 3169 expired.
<0002> tbf.cpp:874 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) T3169 timeout during transsmission
<0002> tbf.cpp:893 - Assignment was on CCCH
<0002> tbf.cpp:899 - No uplink data received yet
<0002> tbf.cpp:879 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) will be freed due to timeout
<0002> tbf.cpp:334 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) free
<0002> bts.cpp:1509 PDCH(TS 6, TRX 0): Detaching TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN), 0 TBFs,
USFs = 00, TFIs = 00000000.
<0002> gprs_ms.cpp:323 Detaching TBF from MS object, TLLI = 0x00000000, TBF = TBF(TFI=0 TLLI=0x00000000 DIR=UL
 STATE=WAIT ASSIGN)
<0002> gprs_ms.cpp:140 Destroying MS object, TLLI = 0x00000000
<0002> tbf.cpp:355 ********** TBF ends here **********
```

Initially, the state is in FLOW until "TX: START Immediate Assignment Uplink (AGCH)"
(according to the commit log, I would have expected ASSIGN, not FLOW here).
Upon the TX, it changes to WAIT_ASSIGN.

It seems we're missing some kind of ack for this "TX: START Immediate Assignment Uplink (AGCH)".

#6 - 06/16/2016 02:53 PM - neels

I can trace the AGCH message from pcu_l1if_tx_agch() in BTS::rcv_rach()
through the PCU socket to l1sap_down() in osmo-bts/src/common/l1sap.c:l1sap_ph_rts_ind().

I'm not familiar enough with the PCU to understand what the expected reply would be.
It's also hard for me to tell where the successful T3169 timer termination would happen.

I'm stopping the investigation now because my search feels unproductive at this point.
Some assistance would be appreciated.

#7 - 06/17/2016 07:32 AM - pierre.baudry

The issue #1742 is about the same error.

As pointed out by Tom and Saurabh on the mailing list, enforcing two-phase-access circumvents the issue.

I believe that the problem is in sched_select_uplink() function in gprs_rlcmac_shed.cpp:

```
static uint8_t sched_select_uplink(uint8_t trx, uint8_t ts, uint32_t fn,
    uint8_t block_nr, struct gprs_rlcmac_pdch *pdch)
{
    struct gprs_rlcmac_ul_tbf *tbf;
    uint8_t usf = 0x07;
    uint8_t i, tfi;

    /* select uplink resource */
    for (i = 0, tfi = pdch->next_ul_tfi; i < 32;
         i++, tfi = (tfi + 1) & 31) {
        tbf = pdch->ul_tbf_by_tfi(tfi);
        /* no TBF for this tfi, go next */
        if (!tbf)
            continue;
        /* no UL resources needed, go next */
        /* we don't need to give resources in FINISHED state,
         * because we have received all blocks and only poll
         * for packet control ack. */
        if (tbf->state_is_not(GPRS_RLCMAC_FLOW))                ///// PROBLEM HERE
            continue;

        /* use this USF */
        usf = tbf->m_usf[ts];
        LOGP(DRLCMACSCHED, LOGL_DEBUG, "Received RTS for PDCH: TRX=%d "
            "TS=%d FN=%d block_nr=%d scheduling USF=%d for "
            "required uplink resource of UL TFI=%d\n", trx, ts, fn,
            block_nr, usf, tfi);
        /* next TBF to handle resource is the next one */
        pdch->next_ul_tfi = (tfi + 1) & 31;
        break;
    }

    return usf;
}
```

#8 - 06/23/2016 09:06 PM - neels

pierre.baudry wrote:

> I believe that the problem is in sched_select_uplink() function in gprs_rlcmac_shed.cpp:
> if (tbf->state_is_not(GPRS_RLCMAC_FLOW)) //// PROBLEM HERE
> continue;

Can you/we explain why this code is not a problem for two phase?

#9 - 06/24/2016 12:02 PM - neels

In https://gerrit.osmocom.org/218 , holger said:

```
I thought the reported already pin-pointed a place where only "flow" is checked and not wait assign?

Did anyone try that simple || state_is(..WAIT_ASSIGN)?
```

Today I checked with this patch, which didn't work but produced other log messages:

```
--- a/src/gprs_rlcmac_sched.cpp
+++ b/src/gprs_rlcmac_sched.cpp
@@ -97,7 +97,8 @@ static uint8_t sched_select_uplink(uint8_t trx, uint8_t ts, uint32_t fn,
                /* we don't need to give resources in FINISHED state,
                 * because we have received all blocks and only poll
                 * for packet control ack. */
-               if (tbf->state_is_not(GPRS_RLCMAC_FLOW))
+               if (tbf->state_is_not(GPRS_RLCMAC_FLOW)
+                   && tbf->state_is_not(GPRS_RLCMAC_WAIT_ASSIGN))
                        continue;

                /* use this USF */
```

The interleaved BTS and PCU log output **without** above patch, i.e. with master d32aa035209364429e8 is:

```
<0007> l1sap.c:920 RACH for packet access
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=123, fn=6216
<0002> tbf.cpp:874 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) T3169 timeout during transsmission
<0002> tbf.cpp:893 - Assignment was on CCCH
<0002> tbf.cpp:899 - No uplink data received yet
, Meas: RSSI -59.21 dBm,  Qual 16.41 dB,  BER 0.00,  Timing -4
<0007> l1sap.c:920 RACH for packet access
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=123, fn=7338
<0002> tbf.cpp:874 TBF(TFI=0 TLLI=0x00000000 DIR=UL STATE=WAIT ASSIGN) T3169 timeout during transsmission
<0002> tbf.cpp:893 - Assignment was on CCCH
<0002> tbf.cpp:899 - No uplink data received yet
, Meas: RSSI -56.70 dBm,  Qual 15.70 dB,  BER 0.00,  Timing -4
<0007> l1sap.c:920 RACH for packet access
[...]
```

The log output **with** above patch changes to:

```
<0007> l1sap.c:920 RACH for packet access
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=120, fn=3054
<0009> tbf_ul.cpp:377 LLC [PCU -> SGSN] TBF(TFI=0 TLLI=0xa00372a3 DIR=UL STATE=WAIT ASSIGN) len=59
<0009> gprs_bssgp_pcu.cpp:182 LLC [SGSN -> PCU] = TLLI: 0xa00372a3 IMSI: 000 len: 9
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0002> tbf.cpp:874 TBF(TFI=0 TLLI=0xa00372a3 DIR=UL STATE=WAIT ASSIGN) T3169 timeout during transsmission
<0002> tbf.cpp:893 - Assignment was on CCCH
<0002> tbf.cpp:897 - Uplink data was received
<0007> gprs_rlcmac_meas.cpp:104 UL RSSI of TLLI=0xa00372a3: -52 dBm
<0009> gprs_bssgp_pcu.cpp:182 LLC [SGSN -> PCU] = TLLI: 0xa00372a3 IMSI: 000 len: 9
<0005> paging.c:254 Add IMM.ASS to queue (group=0)
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
, Meas: RSSI -55.16 dBm,  Qual 17.87 dB,  BER 0.00,  Timing -5
<0007> l1sap.c:920 RACH for packet access
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=120, fn=5298
<0005> tbf.cpp:1154 Got RACH from TLLI=0x00000000 while TBF(TFI=0 TLLI=0xa00372a3 DIR=DL STATE=FLOW) still exi
sts. Killing pending DL TBF
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0009> tbf_ul.cpp:377 LLC [PCU -> SGSN] TBF(TFI=0 TLLI=0xa00372a3 DIR=UL STATE=WAIT ASSIGN) len=17
<0009> gprs_bssgp_pcu.cpp:182 LLC [SGSN -> PCU] = TLLI: 0xa00372a3 IMSI: 000 len: 9
<0005> paging.c:254 Add IMM.ASS to queue (group=0)
<0009> tbf_ul.cpp:377 LLC [PCU -> SGSN] TBF(TFI=0 TLLI=0xa00372a3 DIR=UL STATE=WAIT ASSIGN) len=17
<0009> gprs_bssgp_pcu.cpp:182 LLC [SGSN -> PCU] = TLLI: 0xa00372a3 IMSI: 000 len: 9
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0007> gprs_rlcmac_meas.cpp:184 DL Bandwitdh of IMSI=000 / TLLI=0xa00372a3: 0 KBits/s
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=000 / TLLI=0xa00372a3: 0%
<0002> tbf.cpp:874 TBF(TFI=0 TLLI=0xa00372a3 DIR=UL STATE=WAIT ASSIGN) T3169 timeout during transsmission
<0002> tbf.cpp:893 - Assignment was on CCCH
<0002> tbf.cpp:897 - Uplink data was received
<0007> gprs_rlcmac_meas.cpp:104 UL RSSI of TLLI=0xa00372a3: -52 dBm
<0009> gprs_bssgp_pcu.cpp:182 LLC [SGSN -> PCU] = TLLI: 0xa00372a3 IMSI: 000 len: 9
<0002> tbf.cpp:346 TBF(TFI=0 TLLI=0xa00372a3 DIR=DL STATE=RELEASING) Software error: Pending downlink assignme
nt. This may not happen, because the assignment message never gets transmitted. Please be sure not to free in 
this state. PLEASE FIX!
<0002> tbf.cpp:834 TBF(TFI=1 TLLI=0xa00372a3 DIR=DL STATE=WAIT ASSIGN) releasing due to PACCH assignment timeo
ut.
, Meas: RSSI -55.22 dBm,  Qual 16.70 dB,  BER 0.00,  Timing -5
<0007> l1sap.c:920 RACH for packet access
<0001> pcu_l1_if.cpp:311 RACH request received: sapi=1 qta=0, ra=120, fn=7542
<0009> tbf_ul.cpp:377 LLC [PCU -> SGSN] TBF(TFI=0 TLLI=0xa00372a3 DIR=UL STATE=WAIT ASSIGN) len=17
<0009> gprs_bssgp_pcu.cpp:182 LLC [SGSN -> PCU] = TLLI: 0xa00372a3 IMSI: 274018000000012 len: 26
<0005> paging.c:254 Add IMM.ASS to queue (group=2)
[...]
```

#10 - 06/24/2016 12:03 PM - neels

- Related to Bug #1742: Regression in egprs patch series added

#11 - 06/24/2016 12:06 PM - neels

- Description updated

#12 - 06/24/2016 12:56 PM - neels

- Description updated

#13 - 06/24/2016 01:46 PM - pierre.baudry

neels wrote:

> Can you/we explain why this code is not a problem for two phase?

I haven't investigated on that yet ; I'd bet it's because the MS must send a packet resource request that won't mess with the uplink schedule.

I obtained similar logs with a patch similar to yours.

It seems we never went from the WAIT_ASSIGN state to the FLOW one, even with user traffic flowing.

If I understand one-phase access correctly, we could force the polling bit to '1' (write_ia_rest_uplink) in the Immediate Assignment to request an acknowledgment from the mobile. The TBF should then advance to the FLOW state in 'gprs_rlcmac_pdch::rcv_control_ack()'
If we don't wan't to enforce the packet control acknowledgment, we should find a suitable way to advance to the FLOW state (Hard to figure without knowing the original committer intent).

Thanks for your interest in this issue.

#14 - 06/25/2016 07:29 AM - zecke

neels wrote:

> The log output **with** above patch changes to:
>
> [...]

wow. this looks even worse:

- We get the same timeout
- The MS thinks it timed out too and sends a RACH request
- We kill a pending TBF that is in the state "flow"

Do we fail to stop the/a timer?

#15 - 10/11/2016 09:53 AM - laforge

neels wrote:

> Current osmo-pcu master is not working for GPRS on neither ip.access nanobts nor sysmoBTS.

how is osmo-pcu related to a nanobts?!?

#16 - 10/11/2016 12:14 PM - neels

laforge wrote:

> neels wrote:
>
> > Current osmo-pcu master is not working for GPRS on neither ip.access nanobts nor sysmoBTS.
>
> how is osmo-pcu related to a nanobts?!?

wow, that's clearly a mistake. I probably meant the litecell15...

#17 - 10/11/2016 12:14 PM - neels

- Description updated

#18 - 11/09/2016 10:16 AM - laforge

- Assignee set to neels

wasn't it finally reverted? please make sure to keep issue status up-to-date.

#19 - 11/10/2016 12:40 AM - neels

Indeed, you merged the revert about three weeks ago, which has not come to my attention yet.
Now that you mention it, I notice that this patch is no longer open on gerrit.

I believe the revert is in fact not the right way to go. According to Jacob, the WAIT_ASSIGN
state is the correct way to go (TM), and instead of reverting it, we should fix the bug that
breaks -- let's call it -- non-forced-two-phase-access (i.e. pcu conf without 'two-phase-access')

I am not sure whether anything else is bound to break when we don't have WAIT_ASSIGN,
unfortunately none of us seem familiar enough with the details there.

My last information is that Holger wanted to take a look at it but hasn't found the time yet.

So this issue remains open: undo the revert and fix the problem.
I guess for now we can leave it reverted until we notice a problem that is solved
by the WAIT_ASSIGN state.

#20 - 12/13/2016 10:45 AM - neels

- Priority changed from Normal to Low

#21 - 02/02/2017 04:00 PM - neels

- Assignee changed from neels to Osmocom Developers

#22 - 03/03/2018 09:45 PM - laforge

- Assignee changed from Osmocom Developers to sysmocom