Description
When bootstrapping a Nokia InSite BTS, current OsmoNITB segfaults.

The reason for this is as follows:

- ABM is established.
- LAPD code hands an I frame to the application using send_dl_l3()
- user application decides to call lapd_sap_stop() resulting in a local RELEASE request to LAPD
- LAPD clears the transmit history and changes to IDLE state
- application returns from processing the I frame
- code proceeds in lapd_rx_i() and tries to transmit an I frame, as it didn't realize the state has meanwhile changed
- lapd_send_i() tries to use dl->tx_hist -> boom.

As this is the second bug related to accessing a free’d tx_hist, the code seems to require a more thorough audit.

Related issues:
- Related to libosmocore - Bug #1760: LAPD: segfault in T200 call-back
- Related to libosmocore - Bug #1762: Review LAPD code for race conditions regarding state, particularly in RELEASE
- Related to OsmoBSC - Bug #3975: osmo-bsc crash during startup with nokia insite

History
#1 - 07/03/2016 08:17 PM - laforge
- Related to Bug #1760: LAPD: segfault in T200 call-back added

#2 - 07/03/2016 08:19 PM - laforge
- Status changed from New to In Progress
- % Done changed from 0 to 20

The quick fix for this specific bug is to check for LAPD_STATE_MF_EST in the first lines of labd_send_i(), and return if not. Not sure how many other similar bugs are still hidden :

#3 - 07/03/2016 08:20 PM - laforge
- Related to Bug #1762: Review LAPD code for race conditions regarding state, particularly in RELEASE added

#4 - 11/09/2016 10:20 AM - laforge
- Assignee deleted (laforge)

#5 - 10/05/2017 06:27 AM - laforge
- Status changed from In Progress to New

#6 - 05/08/2019 03:07 PM - laforge
- Assignee set to laforge

#7 - 05/08/2019 03:14 PM - tnt
- Related to Bug #3975: osmo-bsc crash during startup with nokia insite added
I've started to investigate this. Finding a way to solving it is indeed quite tricky so far.

**bts_nokia_site**

lapd_sap\_{start,stop} are called as follows:

- `S_L_INP_LINE_INIT (start OML)`
- `S_L_INP_TEI_UNKNOWN (start RSL)`
- `reset_timer_cbi() (stop all; start OML)`
- when ACK for RESET was received (stop all)
- ACK for CONF_DATA was received (start RSL)

**bts_ericsson_om2000**

lapd_sap\_{start,stop} are called as follows:

- `S_L_INP_LINE_INIT (start)`
- `S_L_INP_LINE_NOALARM (start)`
- `S_L_INP_LINE_ALARM (stop)`

**conclusion so far**

- the critical part is the lapd_sap\_stop()
- it's only critical when used from code paths that will use the SAP afterwards
- input signals should not do this, they are dispatched from driver code
  - `S_L_INP_LINE_INIT` is only generated by e1inp_line\_update() and called from vty
  - `S_L_INP_LINE_ALARM` is currently only generated by DAHDI and called when read/write returns an error or the fd is in exceptfds during select
  - LAPD_ERR\_UNKNOWN\_TEI is generated by LAPD code after all processing

So this means it can currently only be triggered in the Nokia code, and it's likely one of the non-signal cases:

- `reset_timer_cbi() (stop all; start OML)`
  - called from osmocom timer abstraction; ruled out
- ACK for CONF_DATA was received (start RSL)
  - only starts the LAPD link
- when ACK for RESET was received (stop all)
  - this looks like the only code path causing it. We receive an OML message (LAPD I-frame), and during processing of that message we stop the datalink, and then return back into LAPD I-frame processing but the data link is gone.

**#10 - 05/03/2020 09:36 AM - laforge**

fundamentally, this is one of the drawbacks of our 'call everything in-line/synchronous' architecture. This has proven to be suboptimal in a variety of situations already, such as FSM event dispatch, or also here.

If the L3 entity (user of the DL-SAP provided by LAPD) would have an inbound message queue, we would not process the L3 message in-line, but simply put it in the queue and terminate LAPD processing. Once that is finished, we (or some scheduler) would check if there is new data in the L3 input queue, which would process the event. At that point, deleting the LAPD instance was no longer a problem....

**#11 - 05/03/2020 09:40 AM - laforge**

For now the only trivial solution I can see is to consider removing the DL SAP instance during L3 message processing illegal. The Nokia BTS driver must do so asynchronously.

**#12 - 05/03/2020 10:05 AM - laforge**

- Project changed from libosmocore to OsmoBSC
- Category set to Nokia BTS
- % Done changed from 20 to 90

Proposed fix in [https://gerrit.osmocom.org/c/osmo-bsc/+/18009](https://gerrit.osmocom.org/c/osmo-bsc/+/18009) - still needs manual testing/verification