

OsmoBTS - Bug #4281

MS power control loop is interrupted when bad frames are received

11/26/2019 01:25 PM - dexter

Status:	Resolved	Start date:	11/26/2019
Priority:	Normal	Due date:	
Assignee:	dexter	% Done:	100%
Category:			
Target version:			
Spec Reference:			

Description

See <http://git.osmocom.org/osmo-bts/tree/src/common/l1sap.c#n1236>

When a bad frame is received the on l1sap.c, then it is checked if it was a SACCH frame and if it was only radio_link_timeout() is called. The function then exits with -EINVAL. However, later in the code one can find that there is a call to lchan_ms_pwr_ctrl(). We should not skip this step. Instead we should supply the power control loop so that it knows that the reception is not enough and the MS power must be increased.

See also @pespin's comment at <https://gerrit.osmocom.org/c/osmo-bts/+16170/1/src/common/l1sap.c>:

That means in this case lchan_ms_pwr_ctrl() is not called and the MS power loop is missing information. For instance it could mean the strength being used by the MS is not enough and precisely lchan_ms_pwr_ctrl() should be notified so that BTS can instruct the MS to increase the signal strength.

If in this case we don't have block data (data⁰) then MS Power Level value is not available for lchan_ms_pwr_ctrl(). In that case I'd probably makes sense to pass a special value to it, for instance -1, so that it knows the value is not available. Internally it can then for instance use lchan->ms_power_ctrl.current as input for calculations, or simply raise the current MS Power level if rssi is detected to be bad.

History

#1 - 11/26/2019 03:26 PM - dexter

I had a look at int lchan_ms_pwr_ctrl(struct gsm_lchan *lchan, const uint8_t ms_power, const int rxLevel). We could simply call the function with lchan->ms_power_ctrl.current for ms_power (data⁰) and a made up rssi value. That should work.

#2 - 11/26/2019 03:35 PM - pespin

If I understand correctly we still have the correct RSSI value in that case, what we lack is only the MS Power.

I don't know the details about how to decode the data, but it may make sense to separate between 2 cases if possible: 1 byte is available (but not entire block) or not available. In first case we can still feed correct parameters to the function, in the second we cannot.

In the case were no MS power is available we can indeed simply pass lchan->ms_power_ctrl.current.

#3 - 11/26/2019 03:51 PM - dexter

- % Done changed from 0 to 80

I think so, the rssi can be taken from data_ind->rssi. The lchan->ms_power_ctrl.current should be still correct. I do not think that the MS changes its power by itself.

See also: <https://gerrit.osmocom.org/c/osmo-bts/+16233> l1sap.c: ensure ms power control loop is running

#4 - 11/26/2019 03:59 PM - pespin

well lchan->ms_power_ctrl.current is the value we are requesting, but it doesn't necessarily mean it's the one using at that moment:

- The MS may not support the requested ms power level, so it may be transmitting with the nearest supported one (for instance I have an MS which doesn't support power level 29, and in that case transmits with 0, the nearest one below in signal strength).
- We may have just changed lchan->ms_power_ctrl.current from value A to B and MS may still be transmitting with value A or some value between A and B.

So we could discuss about which of the 2 approaches is better:

- Pass `lchan->ms_power_ctrl.current` and let the regular power control loop algo run assuming the tx ms power level is same as `lchan->ms_power_ctrl.current`, meaning output calculated new ms power level is not accurate or wrong.
- Pass special value to tell the algo to simply increase the requested MS power level. However, I'm not sure if that's actually always the correct expected way to solve it, since error to decode could be due to other factors. I'd welcome somebody with more RF knowledge to provide some advise here.

#5 - 11/30/2019 12:10 PM - laforge

On Tue, Nov 26, 2019 at 03:35:39PM +0000, pespin [REDMINE] wrote:

If I understand correctly we still have the correct RSSI value in that case, what we lack is only the MS Power.

ACK.

I don't know the details about how to decode the data, but it may make sense to separate between 2 cases if possible: 1 byte is available (but not entire block) or not available. In first case we can still feed correct parameters to the function, in the second we cannot.

You receive a MAC block and either can fix all bugs and hence the CRC passes, or you have som bit errors, and the CRC fails. If that happens, you have no idea where those bits are.

In the case were no MS power is available we can indeed simply pass `lchan->ms_power_ctrl.current`.

ACK.

#6 - 01/06/2020 10:43 AM - dexter

- *Status changed from New to Resolved*

- *% Done changed from 80 to 100*