OsmoBSC - Bug #3413

TC_lcls_connect_clear does not pass anymore.

07/23/2018 07:39 PM - dexter

Status:	In Progress	**Start date:**	07/23/2018
Priority:	High	**Due date:**	
Assignee:	neels	**% Done:**	90%
Category:			
Target version:			
Spec Reference:			

Description

After we increased the test coverage in #3292 we observed problems with the LCLS tests. (TC_ho_int is a real regression) Those were then fixed, but now TC_lcls_connect_clear indicates problems again. It is not clear yet what causes the failure, but it the location of the problems seems to be in the second interleave after the clear command is sent. The testcase seems to have problems with receiving/matching the tr_RSL_RF_CHAN_REL() template. The test case passes when CONN_A.receive(tr_RSL_RF_CHAN_REL(?)) is removed from the interleave.

When the interactions are traced using wirshark, the RSL CHANNEL RELEASE message can be seen, so it must be somewhere in the template/interleave processing.

Related issues:

Related to OsmoBSC - Bug #3659: LCLS directly between BTSs	**Stalled**	**10/17/2018**

History

#1 - 07/25/2018 02:07 PM - dexter

- File without_tr_RSL_RF_CHAN_REL.log added

- File without_tr_RSL_RF_CHAN_REL.pcapng added

- File with_tr_RSL_RF_CHAN_REL.log added

- File with_tr_RSL_RF_CHAN_REL.pcapng added

- File fail_at_cm_service_req.log added

- File fail_at_cm_service_req.pcapng added

I have now attempted multiple times to fix this without any success. Presumably the problem is a combination of various race conditions in ttcn3. The behavior is unreliable, when I add log statements the behavior changes.

The RF Channel Release is visible in the trace, but appears before the CLEAR COMMAND and the RF CHannel Release Ack can also be seen but it is unclear for me who is sending it since it is even visible when I remove CONN_A.receive(tr_RSL_RF_CHAN_REL(?) from the interleave.

I also tried to tr_RSL_RF_CHAN_REL change so that it gets more tolerant, this did not work either. However, but what is really confusing is that I get a Data Request with a Channel Release that is not acked. Is this not necessary? Even more confusing is that there is indeed an RF Channel Release at the very end of the trace. This one falls in the time frame where the testcase hangs in the interleave.

without_tr_RSL_RF_CHAN_REL.pcapng/log:

```
    /* Perform hard BSSMAP Clear on "A" side, expect no LS on "B" side */
    var myBSSMAP_Cause cause_val := GSM0808_CAUSE_CALL_CONTROL;
    var octetstring l3_rr_chan_rel := '060D00'O;
    CONN_A.send(ts_BSSMAP_ClearCommand(enum2int(cause_val)));

    log("================================ PAST CLEAR CMD ================================");

    interleave {
    [] CONN_A.receive(tr_RSL_DATA_REQ(?, tr_RslLinkID_DCCH(0), l3_rr_chan_rel));
    [] CONN_A.receive(tr_RSL_DEACT_SACCH(?));
//    [] CONN_A.receive(tr_RSL_RF_CHAN_REL(?)) -> value rsl {
//        var RSL_IE_Body ieb;
//        f_rsl_find_ie(rsl, RSL_IE_CHAN_NR, ieb);
//        CONN_A.send(ts_RSL_RF_CHAN_REL_ACK(ieb.chan_nr));
//        }
    [] CONN_A.receive(tr_BSSMAP_ClearComplete) {
```

```
        CONN_A.send(BSSAP_Conn_Prim:MSC_CONN_PRIM_DISC_REQ);
        }
    [] CONN_B.receive(tr_BSSMAP_LclsNotificationSts(LCLS_STS_not_possible_ls));
    }

    log("================================ PAST INTERLEAVE ================================");

    f_sleep(2.0);

    f_lcls_test_fini();
```

with_tr_RSL_RF_CHAN_REL.pcapng/log:

```
    /* Perform hard BSSMAP Clear on "A" side, expect no LS on "B" side */
    var myBSSMAP_Cause cause_val := GSM0808_CAUSE_CALL_CONTROL;
    var octetstring l3_rr_chan_rel := '060D00'O;
    CONN_A.send(ts_BSSMAP_ClearCommand(enum2int(cause_val)));

    log("================================ PAST CLEAR CMD ================================");

    interleave {
    [] CONN_A.receive(tr_RSL_DATA_REQ(?, tr_RslLinkID_DCCH(0), l3_rr_chan_rel));
    [] CONN_A.receive(tr_RSL_DEACT_SACCH(?));
    [] CONN_A.receive(tr_RSL_RF_CHAN_REL(?)) -> value rsl {
        var RSL_IE_Body ieb;
        f_rsl_find_ie(rsl, RSL_IE_CHAN_NR, ieb);
        CONN_A.send(ts_RSL_RF_CHAN_REL_ACK(ieb.chan_nr));
        }
    [] CONN_A.receive(tr_BSSMAP_ClearComplete) {
        CONN_A.send(BSSAP_Conn_Prim:MSC_CONN_PRIM_DISC_REQ);
        }
    [] CONN_B.receive(tr_BSSMAP_LclsNotificationSts(LCLS_STS_not_possible_ls));
    }

    log("================================ PAST INTERLEAVE ================================");

    f_sleep(2.0);

    f_lcls_test_fini();
```

Without the two log statements the things are even worse, then most of the Time we do not even reach the point where we can see the clear command. See fail_at_cm_service_req.log and fail_at_cm_service_req.pcapng for this.

Note: also confusing: In trace without_tr_RSL_RF_CHAN_REL.pcapng packet number 27 RELease REQuest is displayed as malformed packet. The packet seems to come from the BSC. Probably this is a bug. However, I think this should not be the cause of the problem.

#2 - 07/30/2018 10:26 AM - dexter

- Status changed from New to Resolved

The test result analyzer now shows that TC_lcls_connect_clear is passing again. I had a chat with Daniel this morning. He says that he fixed some race conditions in our TTCN3 tests, probably this also fixed the problems of TC_lcls_connect_clear.

Since the test is now passing I set this to resolved.

#3 - 11/01/2018 10:39 AM - msuraev

- Related to Bug #3659: LCLS directly between BTSs added

#4 - 11/01/2018 10:41 AM - msuraev

- Status changed from Resolved to New

- Assignee changed from dexter to neels

This reappeared again on October 12, likely due to one of the commits near 6fe125294b219a519c77f7140de26870d17bf40a in OsmoBSC.

The failing part in TC_lcls_connect_clear is probably CONN_A.receive(tr_RSL_DATA_REQ(?, tr_RslLinkID_DCCH(0), l3_rr_chan_rel)) due to missing "DATA REQuest (DTAP) (RR) Channel Release" packet - see difference in .pcap from https://jenkins.osmocom.org/jenkins/view/TTCN3/job/ttcn3-bsc-test/369/ and https://jenkins.osmocom.org/jenkins/view/TTCN3/job/ttcn3-bsc-test/370/

#5 - 11/01/2018 06:02 PM - neels

As I suspected, I now notice that in BSC_Test.ttcn, usually the channel release process is handled by f_expect_chan_rel(), which does handle various messages **if** they show up, but does not actually verify **which** of these messages are supposed to show up or not.

In the LCLS test, the actual expectation is written out in detail, hence this problem shows up in that TC_lcls_connect_clear(), and not in the various other BSC_Test TCs.

For the normal case, this scenario (MSC sends Clear Command) is covered by BSC_Tests.TC_chan_rel_hard_clear. It is logical that when the MSC requests a Clear, that we do the complete RSL/RR channel release dance, but since f_expect_chan_rel() doesn't care which messages appear, any missing messages there are not caught by the test.

I'm now changing f_expect_chan_rel() to allow indicating whether Deact SACCH and RR Chan Release messages should be present or not, as a basis for fixing the problem in osmo-bsc. I'll also try to make all callers define exactly what they expected to see (which remains optional, but it's better to pinpoint it).

#6 - 11/01/2018 06:09 PM - neels

osmo-bsc commit 6fe125294b219a519c77f7140de26870d17bf40a is not related.
It has an effect only when the RF Chan Rel is sent, i.e. after the RR Release happens.

#7 - 11/06/2018 08:35 PM - neels

- Status changed from New to In Progress

- Priority changed from Normal to High

- % Done changed from 0 to 50

The missing RR Release was introduced by 8b818a01b00ea3daad4ad58c162ac52b4f08a5cb

Got a fix for the missing RR Release: I666b3b4f45706d898d664d380bd0fd2b018be358

And a fix for ttcn3 to properly catch what messages are sent / not sent on release coming up at Ibc64058f1e214bea585f4e8dcb66f3df8ead3845.

#8 - 11/08/2018 05:35 PM - neels

- % Done changed from 50 to 90

Various patches revolving around the RR Release issue are now submitted to gerrit, topic 'rr_release'.
The RR Release fix itself is https://gerrit.osmocom.org/c/osmo-bsc/+/11664 and then there are:

- changes to also send Deact SACCH most of the time.
- changes to ttcn3-bsc-test to catch future errors of this kind.

With all of these patches in, all of osmo-bsc ttcn tests should pass (including TC_lcls_connect_clear)

Files

without_tr_RSL_RF_CHAN_REL.pcapng	6.27 KB	07/25/2018	dexter
without_tr_RSL_RF_CHAN_REL.log	376 KB	07/25/2018	dexter
with_tr_RSL_RF_CHAN_REL.pcapng	11.1 KB	07/25/2018	dexter
with_tr_RSL_RF_CHAN_REL.log	680 KB	07/25/2018	dexter
fail_at_cm_service_req.pcapng	6.27 KB	07/25/2018	dexter
fail_at_cm_service_req.log	373 KB	07/25/2018	dexter