Missing PCU_Tests.ttcn UL TBF tests
04/15/2019 07:54 AM - laforge

Status: New  Start date: 04/15/2019
Priority: High  Due date: 
Assignee: pespin  % Done: 0%
Category: Target version: Spec Reference:

Description
We should have a bunch of automatically executed tests in PCU_Tests.ttcn that cover UL TBF establishment

Related issues:
- Related to OsmoPCU - Bug #3854: OsmoPCU uses wrong CellID in BSSGP  Resolved  03/21/2019
- Related to OsmoPCU - Bug #4180: Regression in osmo-ttcn3-hacks broke f_bssgp_...  Resolved  08/29/2019

History
#1 - 04/15/2019 07:54 AM - laforge
- Priority changed from Normal to High

#2 - 04/15/2019 07:55 AM - laforge
- Tags set to TTCN3

#3 - 08/21/2019 12:47 PM - osmith
- Assignee changed from lynxis to osmith

There are two existing tests:

- TC_ul_tbf
- TC_ul_tbf_single_llc_sizes

https://git.osmocom.org/osmo-ttcn3-hacks/tree/pcu/PCU_Tests.ttcn?id=92c2bdb00fd840ec5fca6e49c12776146b2f4ef4

According to Harald, they were already passing at some point: https://lists.osmocom.org/pipermail/openbsc/2019-July/012992.html

However, we are not running them in jenkins so far and the configs that used to work with the tests are not saved anywhere. I'm trying to get them running again, first locally, then in docker with jenkins for the nightly tests, and then extend the tests from there.

This is the test setup:

```
testsuite ---/tmp/osmocom_l2--- virt_phy ---GSMTAP UDP multicast--- osmo-bts-virtual ---/tmp/pcu_bts--- osmo-pcu
          |   A-bis OML
          | osmo-bsc
```

I've extended my existing ttcn3.sh script to run the above setup locally. WIP branches:

- osmo-ttcn3-hacks.git: osmith/pcu-tests
- osmo-dev.git: osmith/ttcn3.sh

After resolving various issues with my configuration and finding out that the cell_identity in BVCI only gets matched by the testsuite if it is byte-swapped in the config (I'll investigate this further once I have the tests running...related), the current error comes from f_establish_ul_tbf() and says:

```
Timeout establishing UL TBF
```

From reading the test, this should happen:
I'm still investigating where the tune_req/est_req/est_res messages get lost... if somebody has a hint where to look it would be great (fixeria, laforge, lynxis maybe?). I can't find any of them in wireshark, so I'm assuming that virtphy loses them. I've also tried to sniff the first socket with udtrace, but for some reason, during most runs, the ttcn3 dissector fails to output anything (sometimes it had worked though).

#4 - 08/21/2019 12:50 PM - osmith

The timeout does not happen because of any components outside of the testsuite, but because of the code in the testsuite itself, it seems. I've read through the TTCN3 logs again (decided to use the titan eclipse log viewer after a while to make it more feasible), and this is what happens:

```
MTC -> 5: BCCH_tune_req {pcs=false, arfcn=871, combined_ccch=true}
MTC -> 5: TBF_UL_establish_req {tbf_nr=0, ra=123}
MTC <- 5: BCCH_tune_req {pcs=false, arfcn=871, combined_ccch=true, id=1}

5 -> socket: L1CTL_FBSB_REQ {pcs=false, arfcn=871, ...}
5 <- socket: L1CTL_FBSB_CONF {pcs=false, arfcn=871, ...}

MTC <- 5: TBF_UL_establish_req {tbf_nr=0, ra=123, id=2}

5 -> socket: L1CTL_RACH_REQ {ra=123, combined=1, offset=0, ...}
5 <- socket: L1CTL_RACH_CONF
```

I've traced it in the code and found that it is stuck here:
https://git.osmocom.org/osmo-ttcn3-hacks/tree/library/L1CTL_PortType.ttcn?id=92c2bdb00fd840ec5fca6e49c12776146b2f4ef4#n129

After LAPDm_RAW_PT.ttcn:f_establish_tbf() receives L1CTL_FBSB_CONF, it calls f_L1CTL_WAIT_IMM_ASS() to wait for a L1CTL_DATA_IND that has an immediate assignment for the correct RACH.

```
timer T := 10.0;
T.start;
alt {
    [] pt.receive(tr_L1CTL_DATA_IND(t_RslChanNr_PCH_AGCH(0))) -> value dl {
        rr := dec_GsmRrMessage(dl.payload.data_ind.payload);
        log("PCH/AGCH DL RR: ", rr);
        if (match(rr, t_RR_IMM_ASS(ra, rach_fn))) {
            log("Received IMM.ASS for our RACH!");
            return;
        }
    }
    ...
}
```

The "match(rr, t_RR_IMM_ASS(ra, rach_fn))" is always failing, hence it just repeats and never exits this function. The timeout here is set to 10 seconds, but the other timeout that results in "Timeout establishing UL TBF" is only 2 seconds. That is why "Timeout waiting for IMM ASS" is never printed to the log. I'm attaching the current log file ("PCU_Tests.TC_ul_tbf.merged.log") in case somebody wants to take a look at this.

#6 - 08/23/2019 07:28 AM - osmith

I found that this the IMM ASS not getting sent is caused by a change (regression?) in osmo-bts. With old version osmo-bts 1.0.1 it gets past that part.

So I did a bisect. Rebasing on 1.0.1 before the bisect is necessary to have this patch in the tree and therefore prevent build failures:
https://gerrit.osmocom.org/c/osmo-bts/+/13425/

```
$ git checkout master
$ git checkout -b bisect-me
```
$ git rebase 1.0.1
$ git bisect start HEAD 1.0.1
$ git bisect run ~/code/temp/3925/bisect_imm_ass.sh

This is the commit that caused the behavior change:
https://gerrit.osmocom.org/c/osmo-bts/+/14687 ("Move Access Burst link quality handling to L1SAP")

#7 - 08/23/2019 07:47 AM - osmith

The above patch moves the check for link quality. And indeed, the bts log says:

```plaintext
20190823093233821 DL1C <0006> ../../../../src/osmo-bts/src/common/l1sap.c:1279 000881/00/23/14/09 Ignoring RACH request: link quality (0) below the minimum (50)
```

So the testsuite needs to be adjusted to send a proper link quality :)

#8 - 08/23/2019 10:19 AM - osmith

The change needs to be done in osmo-bts actually, and it is a regression from above commit (where the necessary change was made for all other osmo-bts-* variants, but not for virtual). This patch fixes it, and makes the IMM ASS part pass: https://gerrit.osmocom.org/c/osmo-bts/+/15271

#9 - 08/23/2019 10:30 AM - fixeria

Hi Oliver,

This is the commit that caused the behavior change:
https://gerrit.osmocom.org/c/osmo-bts/+/14687 ("Move Access Burst link quality handling to L1SAP")

Sorry, my bad.

So the testsuite needs to be adjusted to send a proper link quality :)

Link quality is defined by C/I ratio. By default, the test cases should be using 60 cB:

https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/14961

```plaintext
[...] Ignoring RACH request: link quality (0) below the minimum (50)
```

It's interesting why the test case is using 0... The recent Wireshark can help you to investigate (dissectors osmo_trxd and osmo_trxc).

#10 - 08/23/2019 10:30 AM - fixeria

Ah, you're not using fake_trx/trxcon, nevermind.

#11 - 08/23/2019 12:57 PM - osmith

Thanks for the input, Vadim!

So with the osmo-bts-virtual fix, these issues are resolved:

- IMM ASS not received
- Timeout establishing UL TBF

TC_ul_tbf is almost completely running through now. The next issue is this one:


The error comes from here in PDU_Tests.ttcn:
function f_bssgp_wait_ul_ud(template PDU_BSSGP exp) runs on dummy_CT {
    timer T := 5.0;
    T.start;
    alt {
        [] BSSGP[0].receive(exp) {
            log("found matching BSSGP UL-UNITDATA PDU");
        }
        [] T.timeout {
            setverdict(fail, "Timeout waiting for ", exp);
            mtc.stop;
        }
    }
}

Wireshark shows this when filtered for bssgp:
...
FLOW-CONTROL-BVC
FLOW-CONTROL-BVC-ACK
SAPI: LLGMM, UI, unprotected, non-ciphered information, N(U) = 0(DTAP)
SAPI: LLGMM, UI, unprotected, non-ciphered information, N(U) = 1(DTAP)
SAPI: LLGMM, UI, unprotected, non-ciphered information, N(U) = 2(DTAP)
SAPI: LLGMM, UI, unprotected, non-ciphered information, N(U) = 3(DTAP)
SAPI: LLGMM, UI, unprotected, non-ciphered information, N(U) = 4(DTAP)
FLOW-CONTROL-BVC
FLOW-CONTROL-BVC-ACK
SAPI: LLGMM, UI, unprotected, non-ciphered information, N(U) = 5(DTAP)

I'll continue to investigate next week.

#12 - 08/26/2019 09:12 AM - osmith
- Checklist item [] Proper fix for byte-swapped cell-id in testsuite added
  - Checklist item [] Proper fix for f_bssgp_wait_ul_ud() added
  - Checklist item [] Get TC_ul_tbf running in jenkins added
  - Checklist item [] Fix running multiple tests after another added
  - Checklist item [] Get TC_paging running in jenkins added
  - Checklist item [] Get TC_selftest_ns running locally added
  - Checklist item [] Get TC_selftest_ns running in jenkins added
  - Checklist item [] Get TC_ul_tbf_single_llc_sizes running locally added
  - Checklist item [] Get TC_ul_tbf_single_llc_sizes running in jenkins added
  - Checklist item [] Develop new tests (using H/L annotation in RAW codec for CSN.1, this checklist item will be split up further) added

The above "Timeout waiting for { pDU_BSSGP_UL_UNITDATA" error was caused by a regression from:
https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/13865 ("BSSGP_Emulation: Abandon "BssgpDecoded" intermediate structure")

After reverting it, I have TC_ul_tbf test passing for the first time :)

So there are four existing tests in total, and now two of them are passing:

- TC_ul_tbf (OK)
- TC_selftest_ns (FAIL)
- TC_ul_tbf_single_llc_sizes (FAIL)
- TC_paging (OK)

Unfortunately, when running multiple tests after another, the tests will also fail (so the shutdown isn't done properly it seems).

I think it makes sense to create proper fixes for the two workarounds I have right now, and then get TC_ul_tbf running in jenkins. That way what I have locally won't divide too much from what we have working in jenkins. From there on, I'll get the two other existing tests running, and then develop new tests. Checklist updated accordingly.

#13 - 08/26/2019 01:36 PM - osmith
- Related to Bug #3854: OsmoPCU uses wrong CellID in BSSGP added

The byte-swapped cell-id is not an issue in the testsuite, but in another component in the chain (likely osmo-bts with code from libosmocore?). See #3854 for details.

#14 - 08/26/2019 01:37 PM - osmith
- Checklist item changed from Proper fix for byte-swapped cell-id in testsuite to Proper fix for byte-swapped cell-id (OS#3854)

The byte-swapped cell-id is not an issue in the testsuite, but in another component in the chain (likely osmo-bts with code from libosmocore?). See #3854 for details.

#15 - 08/29/2019 12:43 PM - osmith
I have gotten the TC_ul_tbf test running docker with jenkins.sh. The biggest obstacle was not having multicast support in docker across multiple containers... but I've worked around that by running virtphy and osmo-bts-virtual in the same container. Pushed to docker-playground.git, osmith/pcu-tests branch. It works, when just running this test.

However, running the existing PCU_Tests_RAW before this test, it gives a "Timeout establishing BSSGP connection" error again. So I will look into proper reset before starting tests next, and when this is resolved, we can run the two working tests in jenkins.

Regarding the proper reset, I've taken a look at the SGSN code (as suggested by lynxis, thanks!) and found that it issues a reset command in f_init_vty:

```
f_vty_transceive(SGSNVTY, "reset sgsn state");
```

I'm trying to run similar reset commands now with the PCU, before running each test. These look promising (1234 would be inserted from the test config):

```
nsvc nsei 1234 reset
nsvc nsvci 1234 reset
```

The PCU tests can't access the OsmoPCU VTY so far, so I've added that first (adjusted the gen_links.sh, regen_makefile.sh etc.). Using the VTY did not work out of the box, because the VTY prompt of OsmoPCU is "Osmo-PCU> " and not "OsmoPCU> ". which does not match the "\w+" pattern used in the TTCN-3 VTY code. After patching it in OsmoPCU, talking to the VTY is working. The reset commands above are not enough it seems, but I'll continue here tomorrow.

I have added a file running-multiple-tests-broken.tar.xz which contains logs and pcaps of running multiple tests after one another:

1. running just TC_ul_tbf (OK)
2. running TC_ul_tbf twice (NOK)
3. running PCU_Tests_RAW.TC_pcuif_suspend, then TC_ul_tbf (similar to how it would run in jenkins) (NOK)

The log file is quite clear:

- the PCU continues "as usual" and sends FLOW-CONTROL-BVC
- there is an ICMP destination unreachable which appears to be ignored by the PCU
  - normally I'd expect this translates to some kind of error behavior on send/recv on the UDP socket? If the PCU would get notified by this, it could BLOCK the NS-VC and BVC and start sending NS_RESET again
- there is no response to the FLOW-CONTROL-BVC
  - the PCU doesn't appear to check if it gets ACKs, or at least not detect the absence of ACKS
  - detecting the absence of ACKs could also be used to BLOCK the NS-VC + BVC and start sending NS_RESET again
- there is no NS_ALIVE procedure within the ~ 20 seconds of the test. This procedure is responsible for checking if the link is still active.
  - not sure what the default timeout is, but I remember those are tuneable, so the PCU config file could probably set a lower timeout, causing NS_ALIVE to be sent e.g. every 10s, which would allow the PCU to detect that the link is down / SGSN is restarted

Quick look at the PCU and libosmobg reveals:
GPRS-NS-ALIVE timer can be configured via VTY.

expiration of NS-ALIVE would trigger S_NS_BLOCK to be dispatched to PCU

PCU would handle S_NS_BLOCK with an internal reset of most of its Gb related state

- however, no NS-RESET would be sent, as the only code path to this is when PCU calls gprs_ns_nsip_connect()

So I guess the PCU should be starting to send NS-RESET in such situations. Basically, there should be a state machine where

- it first sends a NS-RESET, wait for NS-RESET-ACK, and keeps re-transmitting NS-RESET until it receives an ack
- once it has received the RESET-ACK it transitions into ALIVE, BLOCKED and starts the NS-ALIVE procedure
- it then tries to UNBLOCK it and when it's unblocked, it continues with BSSGP bring-up.

If the PCU and/or libosmogb would stick to that proper state machine, I would guess running multiple tests wouldn't be as much of a problem.

What's a bit annoying about all of this is that if using the more modern and spec-compatible IP-SNS (which we only support on the PCU side so far, but not on the SGSN), all of those procedures don't exist anymore anyway. So we could simply chose to run the PCU tests in the IP-SNS dialect of Gb, which should save us from a lot of trouble. But then we still don't have proper functionality of the PCU in the (by now mostly used) non-SNS case.

#23 - 12/05/2019 06:23 PM - pespin

I think this ticket links with my last comment in https://gerrit.osmocom.org/c/osmo-ttpc3-hacks/+/16321:

Do we want to drop old test infra?

#24 - 12/06/2019 07:33 AM - osmith

pespin wrote:

I think this ticket links with my last comment in https://gerrit.osmocom.org/c/osmo-ttpc3-hacks/+/16321:

Do we want to drop old test infra?

As written in gerrit, I think it makes sense.

laforge, what do you think?

#25 - 05/12/2020 12:18 PM - laforge

- Assignee changed from osmith to pespin

pespin please check how much of this is still relevant and change checklist accordingly and/or resolve ticket

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCU_Tests.TC_ul_tbf.merged.log</td>
<td>1.46 MB</td>
<td>08/22/2019</td>
<td>osmith</td>
</tr>
<tr>
<td>running-multiple-tests-broken.tar.xz</td>
<td>1.35 MB</td>
<td>09/05/2019</td>
<td>osmith</td>
</tr>
</tbody>
</table>