OsmoBTS - Feature #3155

execute BTS_Tests.ttcn with real (C123) phone hardware in LTHW setup

04/10/2018 04:28 PM - laforge

Status:	In Progress	**Start date:**	04/10/2018
Priority:	Urgent	**Due date:**	
Assignee:	pespin	**% Done:**	40%
Category:			
Target version:			
Spec Reference:			

Description

We want to execute the BTS_Tests.ttcn against all real hardware BTS models of our LTHW setup.

Hardware Setup

On the hardware side, his means, we need to

- add a C123 with RF cabling to the "modem ports" of the setup
- be able to power cycle this C123 from software
- attach the C123 UART via CP2103 USB-UART to the main unit

Software Setup

We will need to be able to execute the BTS_Tests.ttcn on the gsm tester main unit. This could be done either natively or via the existing docker containers. In any case, we don't want to **build** the ttcn source on the low-power APU, but we want that the compiled test suite is copied/transferred to the APU and then executed there.

We also want to make sure that we have proper locking/exclusivity with the osmo-gsm-tester stack, so that we either run osmo-gsm-tester or BTS_Tests.ttcn.

Finally, this should all be started from jenkins.osmocom.org.

It probably makes sense (as usual) to start with the R&D setup and then replicate the setup in production.

Related issues:

Related to OsmoBTS - Bug #3353: osmo-bts: TTCN3 BTS_Tests.TC_rll_est_ind fail...	**New**	**06/18/2018**
Related to OsmoBTS - Bug #3025: BTS_Tests.TC_paging_tmsi_200percent fails	**Resolved**	**03/02/2018**

History

#1 - 04/10/2018 05:05 PM - laforge

related: https://brmlab.cz/project/gsm/shield

#2 - 04/10/2018 05:42 PM - laforge

https://wiki.cuvoodoo.info/doku.php?id=osmotoserial is the project I was originally referring-to.

We still want to build it into an enclosure, though.

#3 - 04/12/2018 07:31 PM - laforge

- Priority changed from Normal to High

#4 - 04/13/2018 04:51 PM - mschramm

- Status changed from New to In Progress

- % Done changed from 0 to 10

We got three osmotoserial PCBA by tsaitgaist. I got them working in a C123 and in C118 (only a C155 did not come up as expected).

No additional power switch is needed, as the PCBA controls the phone power itself by listening to the state of DTR signal on the USB serial (CP2104).

#5 - 04/13/2018 06:41 PM - mschramm

Only on the C123 the MS-147 plug connects reasonably, but must be fixed somehow too. On at least one C118 used here, it simply falls off. Maybe we strip the phones' enclosures to achieve better contact.

#6 - 04/17/2018 05:20 PM - pespin

List of what's needed after first evaluation:

- HW:
  - A main unit (re-use osmo-gsm-tester main unit)
  - A motorola Cyxz phone plugged with a serial interface plugged in and a software based way to power ir on/off from the main unit.
  - a sysmobts (re-use the osmo-gsm-tester one).
  - RF network setup between motorola phone and sysmobts.
- SW:
  - tools needed to flash and control the motorola Cxyz. Provide either debian packages or a docker image with them. Stuff needed: osmocon, firmware blob, flashing tool?
  - trial borrowed from osmo-gsm-tester build jobs: osmo-bts-sysmo, those will be copied to sysmobts
  - trial borrowed from osmo-gsm-tester build jobs: osmo-bsc. It will configure the sysmobts through OML and tell it to connect via RSL to TTTCN3 test.
  - docker image containing a built and running BTS_Tests.ttcn

Flashing the phone doesn't seem to be required, it can be controlled directly by osmocon through serial at boot time(https://osmocom.org/projects/baseband/wiki/Osmocon)

Then, we need a jenkins job which does the following:

1. Get artifacts from following jobs: osmo-gsm-tester_build-osmo-bts-sysmo osmo-gsm-tester_build-osmo-msc
2. Fetch latest already built docker images
3. run a shell/python script to do all the job.
4. Use the produced junit file.

The script should do the following:

1. Prepare a osmo-bts-sysmo.cfg, osmo-bsc.cfg according to specific setup (ip address, etc., have a look at the one used in BTS_Tests.ttcn)
2. Start osmo-bsc using the prepared osmo-bsc.
3. scp the osmo-bts-trial trial to the sysmobts and start the osmo-bts-sysmo process.
4. Build/download the osmocom firmware.
5. Start osmocon binary, be it in a docker image or via trial.
6. Start latest Bts_Tests ttcn3 docker image, which should start the tests automatically.
7. Wait until Bts_Tests image is done
8. Tear down everything: osmocon, sysmobts processes, osmo-bsc, docker images
9. Store results in artficats and provide a junit file.

Locking mechanism between osmo-gsm-tester and this new run job is not required because it's already integrated in jenkins (the main unit node only executes 1 job in parallel so far).

The job should actually be a matrix job, as we actually want to run the tests for:
- sysmobts
- octphy (requires osmo-bts trial for osmo-bts-octphy)
- sysmocell500 (requires osmo-bts trial for osmo-bts-trx)
- nanobts

Remark:
It may actually make sense to reuse osmo-gsm-tester to start and manage all those processes (BTS+BSC), and provide a specific test which waits for a specific event in the system to finish (for instance a unix socket receiving some stuff).
For instance the shell script starting osmo-gsm-tester with that mentioned test, then starting osmocon, then starting the docker image with the ttcn3 tests.

#7 - 04/17/2018 08:09 PM - laforge

On Tue, Apr 17, 2018 at 05:20:41PM +0000, pespin [REDMINE] wrote:

> List of what's needed after first evaluation:
>
> - HW:
>   - A main unit (re-use osmo-gsm-tester main unit)
>   - A motorola Cyxz phone plugged with a serial interface plugged in and a software based way to power ir on/off from the main unit.

let's focus on C123 or C118, and make sure it's the same model everywhere. I can bring
more phones from home, as needed.

> - SW:
>   - tools needed to flash and control the motorola Cxyz. Provide either debian packages or a docker image with them. Stuff needed: osmocon, firmware blob, flashing tool?

there is no "flashing" involved, it's all just loaded into ram. You need the following bits
and pieces from osmocom-bb.git:

- osmocon
- layer1 firmware image (cross-compiled with ancient toolchain)
  - make sure to enable transmit in the firmware Makefile!

- docker image containing a built and running BTS_Tests.ttcn

It still remains to be seen if the docker image can be executed on the APU (resource wise)

We clearly don't have to build the docker image on the APU itself, but we have to run it there.

> Flashing the phone doesn't seem to be required, it can be controlled directly by osmocon through serial at boot time(https://osmocom.org/projects/baseband/wiki/Osmocon)

ACK.

> 1. Fetch latest already built docker images

we can push those from the build hosts to our own registry (We do have a container running on
admin2.sysmocom.de, but it's not used/tested yet)

> 1. Build/download the osmocom firmware.

I think it's fine to build it on jenkins build slaves (I think we even have an osmocom-bb build verification job already,
it's just not storing the resulting builds and probably builds without TX enabled)

> 1. Start latest Bts_Tests ttcn3 docker image, which should start the tests automatically.

correct, if the phone, BTS, BSC and 'osmocon' are all running, executing the tests works exactly
like in the virtualized environment with trxcon+fake_trx.

#8 - 04/18/2018 06:26 PM - mschramm

- % Done changed from 10 to 20

A phone is now installed with an osmotoserial PCBA in the R&D test rig. However, it switches on immediately, though no terminal process is working for this serial (right now it is /dev/ttyUSB14) which is different to the behaviour seen on my local machine. I'll leave this to pespin; because of OsmoDevCon I don't think that progress on this will be made before upcoming week.

#9 - 05/17/2018 11:39 AM - laforge

pespin: What's the status here? I think this should be done still in may, so that you won't have to touch the hardware setup after your move from Berlin.

#10 - 05/17/2018 11:39 AM - laforge

- Priority changed from High to Urgent

#11 - 05/18/2018 03:35 PM - pespin

- % Done changed from 20 to 30

I finally decided to run TTCN3 binaries from inside osmo-gsm-tester test, once everything is set up.

Work done so far can be found in: https://git.osmocom.org/osmo-gsm-tester/log/?h=pespin/ttcn3
I already have osmocon built (firwmare missing) and a class to use it in osmo-gsm-tester. I have a test using the osmocon class, starting all the required elements (BTS, BSC, etc.) and which calls a script to start the docker once everything is set up. When docker finishes, everything is teared down.

For the missing parts, here's a TODO list:

```
TODO:
* Build osmocom-bb's firwmare (cross-compile) in jenkins job build.
* Find a way to programatically find /dev/ttyUSBX for the motorla phone.
* Create script which fetches latest ttcn3-hacks docker image and starts osmo-gsm-tester ttcn3 test.
** status: we need to build the "ttcn3-bts-test" docker container and pull it in osmo-gsm-tester
* ansible: Add "docker" to ansible osmo-gsm-tester buildhost.
** status: Code to do it already exists in our ansible files. We nedd to apply it to osmo-gsm-tester nodes.
```

#12 - 05/19/2018 10:40 AM - laforge

On Fri, May 18, 2018 at 03:35:31PM +0000, pespin [REDMINE] wrote:

- Find a way to programatically find /dev/ttyUSBX for the motorla phone.

if you use a sysmocom CP2102 adapter, it will have a device-unique fixed serial number,
which you can find in /dev/serial/by-id/

#13 - 05/23/2018 06:34 PM - pespin

Current state:

I have most of the required elements set up and running:
- TTCN3 BTS tests are pulled from registry.sysmocom.de docker.
- jenkins job to run the tests in prod setup (+ merged in osmo-ci):
https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_ttcn3/
- improvmenets of osmo-gsm-tester + specific ttcn3 test subdir with all required stuff to run the TTCN3 tests. Can be found in
https://git.osmocom.org/osmo-gsm-tester/log/?h=pespin/ttcn3
- osmocon with motorola phone seems to be running fine in in RnD main unit.
- A BTS_Tests.cfg is generated from a template with the IP addr used by the BTS set up by osmo-gsm-tester.
- BTS is respawned automatically when it stops (due to TTCN3 tests closing the RSL link).
- docker container is killed correctly if osmo-gsm-tester test times out.
- I can run the setup in RnD through ssh easily: OSMO_GSM_TESTER_CONF=/home/pespin/osmo-gsm-tester/ttcn3 ./run_remote.sh -T -l dbg
- All TTCN3 tests are stored correctly under osmo-gsm-tester test's rundir, ready to be packer and stored as an artifact.

Current status:
There seem to be some issues with TTCN3 connecting to IPA-CTRL, I'll have to look at the logs+pcap files for more details.

TODO:
- Add some manually installed stuff in main unit with ansible
- Collect the TTCN3 junit log and make sure the run.tar.gz is stored as artifact.

#14 - 05/24/2018 04:42 PM - pespin

Updated status:

- All elements are in place and starting/stopping correctly in https://git.osmocom.org/osmo-gsm-tester/log/?h=pespin/ttcn3
- I fixed the TTCN3 IPA-CTRL reported issues, now some tests are already passing.
- Required config in ansible are already in gerrit and setup correctly in https://gerrit.osmocom.org/#/c/osmo-ci/+/9274/
- TTCN3 junit log is collected in the correct path.

TODO:

- Add motorola phone to Prod setup (with roh) so we can run the jenkins setup.
- Investigate issue with osmocon/kernel/hw not switching the DTR correctly most of the times in the RnD main unit (with mschramm).
- TTCN3 has an issue opening the PCU socket, but the file is there in docker (I checked with file /data/unix/pcu_sock). I added a commit in ttcn3 to print the error str associated in https://gerrit.osmocom.org/#/c/osmo-ttcn3-hacks/+/9273/. Once it's merged, I need to push an updated ttcn3-bts-test docker image and re-try and see the reason. For now I disabled the PCU socket in the tests by using an empty string as pcu socket path in BTS_Tests.cfg.
- Most tests still don't pass, they give some errors probably related to l1 after getting into osmocon. I still need to check into the details.

#15 - 05/25/2018 04:02 PM - pespin

a motorola phone has been installed in the prod setup, but it seems it has the same serial issues than the one in RnD setup (phone not power cycled when osmocon process is started and serial device is opened).

#16 - 05/25/2018 07:10 PM - laforge

On Fri, May 25, 2018 at 04:02:18PM +0000, pespin [REDMINE] wrote:

> a motorola phone has been installed in the prod setup, but it seems it has the same serial issues than the one in RnD setup (phone not power cycled when osmocon process is started and serial device is opened).

as it has been mentioned before, it's not the phone that has the issue, but it's the operating system (most likely cp210x usb-serial driver) in debian stable.

Have you meanwhile tried to adjust DTR manually? Maybe it's just the automatic DTR switch-off (high-level) on port close that's broken, and the manuel setting via termios still works? In that case, a small modification in osmocon might be sufficient.

If that also doeesn't work, I suppose we'll have to resort to building a patched kernel, or at very least a patched cp210x.ko :/ It may also make sense to report this to Debian upstream so they can fix it in their next stable kernel update?

#17 - 05/25/2018 09:10 PM - pespin

laforge wrote:

> Have you meanwhile tried to adjust DTR manually? Maybe it's just the automatic DTR switch-off (high-level) on port close that's broken, and the manuel setting via termios still works? In that case, a small modification in osmocon might be sufficient.

The manual setting of DTR is already being done at startup.
osmocon calls "osmo_serial_init" in https://git.osmocom.org/osmocom-bb/tree/src/host/osmocon/osmocon.c#n1467
Then osmo_serial_init sets DTR: https://git.osmocom.org/libosmocore/tree/src/serial.c#n112

So if I understand correctly, your idea is to try calling ioctl TIOCMBIC (clear bit) TIOCM_DTR before calling close() on the fd right? I'll give it a try on Monday.

#18 - 05/26/2018 07:50 AM - laforge

On Fri, May 25, 2018 at 09:10:53PM +0000, pespin [REDMINE] wrote:

> laforge wrote:
>
> > Have you meanwhile tried to adjust DTR manually? Maybe it's just the automatic DTR switch-off (high-level) on port close that's broken, and the manuel setting via termios still works? In that case, a small modification in osmocon might be sufficient.
>
> The manual setting of DTR is already being done at startup.

yes, that's clear. But AFAIK the problem was not that the phone isn't powered on, but it is that it's never powered off.

> So if I understand correctly, your idea is to try calling ioctl TIOCMBIC (clear bit) TIOCM_DTR before calling close() on the fd right? I'll give it a try on Monday.

I can't recite the details about the ioctl() to use, but yes, the idea is to explicitly release DTR before close. Or, in case of an unclean shutdown, one might also think of a DTR cycling at startup (i.e. first clear/release DTR, wait a second, then set it).

#19 - 05/26/2018 08:39 AM - laforge

- File cp210x-dkms_4.16-0.tar.gz added

- File cp210x-dkms_4.16-0.dsc added

- File cp210x-4.16.tar.bz2 added

- File cp210x-dkms_4.16-0_all.deb added

I created a DKMS package for current mainline cp210x which can be build on debian 9. I also created a debian package that can be installed with dpkg -i and which will ensure the module will be re-build using dkms on any future kernel update.

Please try if this fixes the problem, and if so, integrate with ansible.

#20 - 05/28/2018 12:43 PM - pespin

I did several tests by clearing the DTR bits, then setting them after receiving the fd from osmo_serial_init():

```
dnload.serial_fd.fd = osmo_serial_init(serial_dev, MODEM_BAUDRATE);
+       /* Reset ready to read/write */
+       fprintf(stderr, "Clearing DTR bit\n");
+       int v24 = TIOCM_DTR | TIOCM_RTS;
+       if (ioctl(dnload.serial_fd.fd, TIOCMBIC, &v24) < 0)
+               fprintf(stderr, "ioctl(TIOCMBIC) %d", errno);
+       usleep(5*1000*1000);
+       fprintf(stderr, "Setting DTR bit\n");
+       if (ioctl(dnload.serial_fd.fd, TIOCMBIS, &v24) < 0)
+               fprintf(stderr, "ioctl(TIOCMBIS) %d", errno);
```

I also tried using getchar() instead of usleep() to control the state changes and be able to visually check the modem.

First times I run it, it seemed to have to difference: Modem still prints "Charinging" in the screen, and I get nothing in the serial (osmocon blocked in select()).
I then re-plugged the usb connector while osmocon was running (with DTR suppousedly on), and then next time I run osmocon the fw was loaded correctly. But then next times I saw that actually the previously loaded by osmocon firmware was still running, and that the modem was not being rebooted. So it is now running the osmocom fw even when the serial session is closed, or even when we explicitly clear the DTR bit as shown by code above.

So it really seems kernel is not correctly setting the DTR bit on the device. I'll give a try to the dkms package and see how it goes.

#21 - 05/28/2018 01:56 PM - pespin

I tested the provided kernel module, and I got same results, it is still not working.

I used:

```
rmmod cp210x
dpkg -i ....deb
modprobe cp210x
```

And then checked that the out of tree module was loaded:

```
[868127.383269] cp210x: loading out-of-tree module taints kernel.
```

#22 - 05/28/2018 02:51 PM - pespin

I'm testing now the motorola phone in an APU2 I have on my table, running same debian and kernel version than the one in osmo-gsm-tester RnD setup:

```
Linux apu 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07) x86_64 GNU/Linux
```

I connected the usb serial connector directly to one of the USB slots in the APU2 board. Everything seems to be working fine with osmocon, like in my PC.

#23 - 05/28/2018 03:10 PM - laforge

On Mon, May 28, 2018 at 02:51:14PM +0000, pespin [REDMINE] wrote:

> I'm testing now the motorola phone in an APU2 I have on my table, running same debian and kernel version than the one in osmo-gsm-tester RnD setup:
>
> I connected the usb serial connector directly to one of the USB slots in the APU2 board. Everything seems to be working fine with osmocon, like in my PC.

this is very odd.

Maybe there is some other process on the gsm-tester APU which open the serial port [in parallel]?

If there's of course another process holding the port still open, then existing 'osmocon' will
not release DTR.

'lsof' should help to find out...

#24 - 05/28/2018 03:27 PM - pespin

I finally found how to trigger the work and non-working state for the motorola phone

- plug in the RF antenna -> screen prints "charging" and osmocon cannot connect
- unplug the RT antenna -> screen stops printing "charging" and osmocon can connect

#25 - 05/29/2018 03:13 PM - roh

i reworked 2 boards to use high-side switching by using AAT4626 from our lab-stock (any high side switch would do)

since they expect high level i removed just the power-fet, bridged ground and cut a trace to have the vbat (via diode) and vcharge switched and the pullups not switched. works nicely for me (tested with neocon and minicom)

the boards are connected to the -dev and -prod testers again.

#26 - 05/29/2018 03:58 PM - pespin

I confirm -dev can be opened with osmocon fine all the time now.
For prod, it seems minicom powers on/off the device fine, but I don't know why yet, osmocon doesn't power it on. I tried updating the cp210x module to the 4.16 dkms one, like in -dev, but it didn't help with the issue.

#27 - 05/30/2018 02:20 PM - laforge

- Tags set to TTCN3

#28 - 05/31/2018 09:47 AM - pespin

I rebooted the machine and made sure to start with a clean libosmocore+osmocon setup to investigate why was the serial not working properly yet, but it turns out after using this clean test env it works fine now, like in RnD. So not sure why, but it's working now. I'm starting to run the jenkins job to get some first results and see how far can I get.

#29 - 05/31/2018 12:21 PM - pespin

First jenkins run finished, report can be found in:
https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_ttcn3/21/testReport/

66 failures out of 84 tests.

The archived artifacts should contain all the information required to find out what's failing in each case.
Some interesting files/paths:

- osmocon log: run.2018-05-31_11-41-09/ttcn3_bts_tests:trx/ttcn3_bts_tests.py/osmocon_ttyUSB3/stdout
- TTCN3 log: run.2018-05-31_11-41-09/ttcn3_bts_tests:trx/ttcn3_bts_tests.py/ttcn3/stdout
- BTS_Tests.cfg used: run.2018-05-31_11-41-09/ttcn3_bts_tests:trx/ttcn3_bts_tests.py/ttcn3/BTS_Tests.cfg
- Separate test logs + pcaps: run.2018-05-31_11-41-09/ttcn3_bts_tests:trx/ttcn3_bts_tests.py/ttcn3/logs/bts-tester/

#30 - 05/31/2018 12:51 PM - laforge

On Thu, May 31, 2018 at 12:21:27PM +0000, pespin [REDMINE] wrote:

> 66 failures out of 84 tests.

this is **highly** unexpected. As indicated before, I was able to get almost all tests to pass, except those dealing with measurement values, which is of course expected.

From looking at the results, I would expect at least the following to pass:

- TC_chan_act_*
- TC_deact_sacch
- TC_encr_cmd_*
- TC_rll_*
- TC_rll_*
- TC_sacch_*
- TC_si_sched_*

All of the above have nothing related to measurement or statistics/rate, and should pass.

Even more worrying are

- TC_dyn_ipa_pdch_act_deact
- TC_dyn_ipa_pdch_act_tchf_act_nack
- TC_dyn_osmo_pdch_act_deact
- TC_dyn_osmo_pdch_double_act
- TC_pcu_act_req
- TC_pcu_act_req_wrong_bts
- TC_pcu_act_req_wrong_trx
- TC_pcu_act_req_wrong_ts
- TC_pcu_deact_req
- TC_pcu_deact_req_wrong_ts

which all don't even use L1CTL / the phone. They should work even without any phone present in the first place, as they only use the IP based interfaces of the BTS to test.

So I think there's still a lot going wrong in the setup.

In fact, I think from the tests results you are getting, not a single one that passes is using the phone.

So we have

- tests that don't use the phone but don't pass (very odd)
- tests that don't use the phone but pass (good)
- tests that use the phone all don't pass

#31 - 06/07/2018 05:55 PM - pespin

I submited patch https://gerrit.osmocom.org/#/c/9492/ in order to change the RxLev sent through L1CTL, but it seems it's not needed so far and actually the arfcn sent by ttcn3 was not matching the one set up by osmo-gsm-tester. By configuring the arfcn correctly in the BTS_Tests.cfg.tmpl file tests now reach further and ttcn3 states it receives Assignment Commands after sending a RACH.

However, at some point both ends (ttcn3 and osmocon) started failing stating an error with the unix socket. ttcn3 "connection refused" and osmocon "Connection reset by peer".

osmocon: Err from socket: Connection reset by peer
ttcn3: TC_sacch_info_mod(92)@64afac245c1c: Dynamic test case error: Stand-alone receive statement failed in file L1CTL_PortType.ttcn, line 190. (Connection refused)

It seems ttcn3 test is probably re-connecting the unix socket in f_connect_reset(), and osmocon doesn't like that and exits. This behavior was not shown while Harald did the BTS tests without real hardware because in there trxcon is used instead of osmocon, so it seems osmocon needs to be fixed/improved to support this kind of scenarios.

About the "tests that don't use the phone but don't pass" from Harald's last post, I bet it's related to the fact that I disabled using the PCU socket (BTS_Tests.mp_pcu_socket := "") due to some issues I was finding in the setup, I'll look into that later.

#32 - 06/07/2018 07:08 PM - fixeria

Hi Pau,

I think that the problem is here:

https://git.osmocom.org/osmocom-bb/tree/src/host/osmocon/osmocon.c#n1161

it exits if 'rc == 0', while trxcon doesn't exit in such cases.

I hope, this will help.

#33 - 06/12/2018 01:43 PM - pespin

I tried removing that exit path (rc==0) and added an extra stderr log line to show what read() from handle_buffer returns.

```
commit a607471194aab81261c9b67a41a871eb36a42e6e (HEAD -> pespin/no-exit, origin/pespin/no-exit)
Author: Pau Espin Pedrol <pespin@sysmocom.de>
Date:   Tue Jun 12 14:38:28 2018 +0200

    osmocon: Do not exit on read return 0

    Change-Id: Ifedfbc9742792b4f8d7850dc7ce84d5b3dbc0096

diff --git a/src/host/osmocon/osmocon.c b/src/host/osmocon/osmocon.c
```

```
index 76f60374..e9427884 100644
--- a/src/host/osmocon/osmocon.c
+++ b/src/host/osmocon/osmocon.c
@@ -789,8 +789,11 @@ static int handle_buffer(int buf_used_len)
        }

        nbytes = read(dnload.serial_fd.fd, bufptr, buf_left);
-       if (nbytes <= 0)
+       if (nbytes <= 0) {
+               fprintf(stderr, "handle_buffer() read(%d) = %d (%s)\n",
+                       buf_left, nbytes, strerror(errno));
                return nbytes;
+       }

        if (!dnload.expect_hdlc) {
                printf("got %i bytes from modem, ", nbytes);
@@ -1173,8 +1176,6 @@ static int serial_read(struct osmo_fd *fd, unsigned int flags)
                                while ((rc = handle_read()) > 0);
                                break;
                }
-               if (rc == 0)
-                       exit(2);
        }

        if (flags & BSC_FD_WRITE) {
```

It seems however that osmocon is not finishing in there. It seems to exit due to a SIGPIPE error. I could get this trace while running it out of osmo-gsm-tester under gdb while running the TTCN3 tests:

```
handle_buffer() read(1) = -1 (Resource temporarily unavailable)
handle_buffer() read(1) = -1 (Resource temporarily unavailable)
handle_buffer() read(1) = -1 (Resource temporarily unavailable)
handle_buffer() read(1) = -1 (Resource temporarily unavailable)

Program received signal SIGPIPE, Broken pipe.
0x00007ffff76da730 in __write_nocancel () at ../sysdeps/unix/syscall-template.S:84
84      ../sysdeps/unix/syscall-template.S: No such file or directory.
(gdb) bt
#0  0x00007ffff76da730 in __write_nocancel () at ../sysdeps/unix/syscall-template.S:84
#1  0x0000555555556e31 in hdlc_tool_cb (dlci=5 '\005', msg=0x55555576cec0) at /home/pespin/dev/git/osmocom-bb/
src/host/osmocon/osmocon.c:769
#2  0x0000555555559376 in dispatch_rx_msg (dlci=5 '\005', msg=0x55555576cec0)
    at /home/pespin/dev/git/osmocom-bb/src/host/osmocon/../../target/firmware/comm/sercomm.c:243
#3  0x000055555555949c in sercomm_drv_rx_char (ch=126 '~') at /home/pespin/dev/git/osmocom-bb/src/host/osmocon
/../../target/firmware/comm/sercomm.c:286
#4  0x0000555555556fec in handle_buffer (buf_used_len=7) at /home/pespin/dev/git/osmocom-bb/src/host/osmocon/o
smocon.c:804
#5  0x000055555555703b in handle_read () at /home/pespin/dev/git/osmocom-bb/src/host/osmocon/osmocon.c:816
#6  0x0000555555557d56 in serial_read (fd=0x55555575c358 <dnload+24>, flags=1) at /home/pespin/dev/git/osmocom
-bb/src/host/osmocon/osmocon.c:1176
#7  0x00007ffff79a83ae in osmo_fd_disp_fds (_eset=0x7fffffffe2a0, _wset=0x7fffffffe220, _rset=0x7fffffffe1a0)
at select.c:217
#8  osmo_select_main (polling=<optimized out>) at select.c:257
#9  0x00005555555588df in main (argc=10, argv=0x7fffffffe478) at /home/pespin/dev/git/osmocom-bb/src/host/osmo
con/osmocon.c:1523
```

#34 - 06/12/2018 01:44 PM - pespin

It's strange too that I see a lot of EAGAIN errors in read(), because afaik it should be driven by select(), so it should only signal READ cb if there's something to read from the device...

#35 - 06/12/2018 02:55 PM - pespin

https://gerrit.osmocom.org/#/c/osmocom-bb/+/9562 seems to fix the premature exit of osmocon while running ttcn3 BTS_Tests.

#36 - 06/12/2018 03:18 PM - pespin

- % Done changed from 30 to 40

Running with trx-b200 in RnD setup after the patch shared in last comment:

```
Comparing expected results /osmo-ttcn3-hacks/bts/expected-results.xml against results in junit-xml-19.log
--------------------
pass BTS_Tests.TC_chan_act_stress
pass BTS_Tests.TC_chan_act_react
pass BTS_Tests.TC_chan_deact_not_active
pass BTS_Tests.TC_chan_act_wrong_nr
pass->FAIL BTS_Tests.TC_deact_sacch
pass BTS_Tests.TC_sacch_filling
pass BTS_Tests.TC_sacch_info_mod
pass BTS_Tests.TC_sacch_multi
xfail BTS_Tests.TC_sacch_multi_chg
pass BTS_Tests.TC_rach_content
pass->FAIL BTS_Tests.TC_rach_count
pass->FAIL BTS_Tests.TC_rach_max_ta
pass->FAIL BTS_Tests.TC_meas_res_sign_tchf
pass->FAIL BTS_Tests.TC_meas_res_sign_tchh
pass BTS_Tests.TC_meas_res_sign_sdcch4
pass BTS_Tests.TC_meas_res_sign_sdcch8
pass->FAIL BTS_Tests.TC_meas_res_sign_tchh_toa256
pass->FAIL BTS_Tests.TC_conn_fail_crit
pass->FAIL BTS_Tests.TC_paging_imsi_80percent
pass->FAIL BTS_Tests.TC_paging_tmsi_80percent
pass->FAIL BTS_Tests.TC_paging_imsi_200percent
pass->FAIL BTS_Tests.TC_paging_tmsi_200percent
pass BTS_Tests.TC_rsl_protocol_error
pass BTS_Tests.TC_rsl_mand_ie_error
pass BTS_Tests.TC_rsl_ie_content_error
pass BTS_Tests.TC_si_sched_default
pass BTS_Tests.TC_si_sched_1
pass BTS_Tests.TC_si_sched_2bis
pass BTS_Tests.TC_si_sched_2ter
pass BTS_Tests.TC_si_sched_2ter_2bis
pass->FAIL BTS_Tests.TC_si_sched_2quater
pass BTS_Tests.TC_si_sched_13
pass->FAIL BTS_Tests.TC_si_sched_13_2bis_2ter_2quater
pass BTS_Tests.TC_ipa_dlcx_not_active
pass BTS_Tests.TC_ipa_crcx_twice_not_active
pass BTS_Tests.TC_ipa_crcx_mdcx_dlcx_not_active
pass BTS_Tests.TC_ipa_crcx_mdcx_mdcx_dlcx_not_active
pass BTS_Tests.TC_ipa_crcx_sdcch_not_active
pass->FAIL BTS_Tests.TC_pcu_act_req
pass->FAIL BTS_Tests.TC_pcu_act_req_wrong_ts
pass->FAIL BTS_Tests.TC_pcu_act_req_wrong_bts
pass->FAIL BTS_Tests.TC_pcu_act_req_wrong_trx
pass->FAIL BTS_Tests.TC_pcu_deact_req
pass->FAIL BTS_Tests.TC_pcu_deact_req_wrong_ts
pass->FAIL BTS_Tests.TC_pcu_ver_si13
pass->FAIL BTS_Tests.TC_pcu_data_req_wrong_bts
pass->FAIL BTS_Tests.TC_pcu_data_req_wrong_trx
pass->FAIL BTS_Tests.TC_pcu_data_req_wrong_ts
xfail BTS_Tests.TC_pcu_data_req_ts_inactive
pass->FAIL BTS_Tests.TC_pcu_data_req_pdtch
pass->FAIL BTS_Tests.TC_pcu_data_req_ptcch
pass->FAIL BTS_Tests.TC_pcu_data_req_agch
```

```
pass->FAIL BTS_Tests.TC_pcu_data_req_imm_ass_pch
pass->FAIL BTS_Tests.TC_pcu_rach_content
pass->FAIL BTS_Tests.TC_pcu_paging_from_rsl
pass->FAIL BTS_Tests.TC_dyn_osmo_pdch_act_deact
pass BTS_Tests.TC_dyn_osmo_pdch_unsol_deact
pass->FAIL BTS_Tests.TC_dyn_osmo_pdch_double_act
pass BTS_Tests.TC_dyn_osmo_pdch_tchf_act
pass BTS_Tests.TC_dyn_osmo_pdch_tchh_act
pass->FAIL BTS_Tests.TC_dyn_ipa_pdch_act_deact
pass BTS_Tests.TC_dyn_ipa_pdch_tchf_act
pass BTS_Tests.TC_dyn_ipa_pdch_tchf_act_pdch_act_nack
pass->FAIL BTS_Tests.TC_dyn_ipa_pdch_act_tchf_act_nack
pass->FAIL BTS_Tests.TC_rll_est_ind
pass BTS_Tests.TC_rll_est_req_DCCH_3
pass BTS_Tests.TC_rll_est_req_ACCH_3
pass->FAIL BTS_Tests.TC_rll_rel_ind_DCCH_0
pass->FAIL BTS_Tests.TC_rll_rel_ind_DCCH_3
pass BTS_Tests.TC_rll_rel_ind_ACCH_0
pass BTS_Tests.TC_rll_rel_ind_ACCH_3
pass->FAIL BTS_Tests.TC_rll_rel_req
pass BTS_Tests.TC_rll_unit_data_req_DCCH
pass BTS_Tests.TC_rll_unit_data_req_ACCH
pass->FAIL BTS_Tests.TC_rll_unit_data_ind_DCCH
pass BTS_Tests.TC_rll_unit_data_ind_ACCH
pass BTS_Tests.TC_chan_act_a51
pass BTS_Tests.TC_chan_act_a52
pass->FAIL BTS_Tests.TC_chan_act_a53
pass->FAIL BTS_Tests.TC_encr_cmd_a51
```

So around half of them are passing now.

The PCU ones probably fail because I disabled passing the PCU socket path, I need to investigate why it couldn't open it to enable it again.

#37 - 06/13/2018 03:05 PM - pespin

junit jenkins output can already be found in

https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_ttcn3/test_results_analyzer/

As mentioned before, I still need to check the PCU socket issue.

#38 - 06/15/2018 03:41 PM - pespin

PCU socket issue fixed (by bind mounting the directory instead of the file, otherwise if the file is recreated outside the container, the container cannot see the new file and keeps seeing the old one).

I modified the code to store the XML junit file as change its classname to include the osmo-gsm-tester suite name, this way if we have for instance:
ttcn3_bts_tests:trx-b200
ttcn3_bts_tests:trx-sysmo5k
ttcn3_bts_tests:sysmo

We get 3 xml files, each of them having a different class name and all of them end up being shown in Jenkins' "Test Results Analyzer".

All the required bits can be found in 1 commit in osmo-gsm-tester branch "pespin/ttcn3".

TTCN3 running with a trx-b200 can be found in:
https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_ttcn3/test_results_analyzer/

I need to find a way to clean the build history of the job, otherwise the old entries (without having the osmo-gsm-tester suite name) still are shown and it's a bit messy.

#39 - 06/16/2018 03:10 PM - laforge

There are still way too many tests failing, so I suspect there's still some fundamental
problem. The TC_rll_* and TC_sacch_* should be exactly like in the virtual setup.

I still believe it is best to establish a baseline with a OsmocomBB phone + BTS
on your desk and then compare that to the automatic test setup.

Regards,
Harald

#40 - 06/17/2018 10:10 PM - pespin

I see a lot of tests (rll and sacch) failing with this kind of message:

"Dynamic test case error: Error message was received from MC: The connect operation refers to test component with component reference 41, which has already terminated."

I can now easily run separate tests in the RnD setup through ssh, so I can debug each test in there. As far as I remember, I noticed that some tests observed to be failing in the Prod automated run (Test Results Analyzer) passed fine when running them separately in RnD.
I'm still not applying the Rxlev != 57 in the tests, since they seemed to work fine enough with 57 as used in the virt setup. I know half of the tests are not yet passing, but after fixing the PCU part I wanted to let them run for a while to see the success/fail patterns and see what kind of error to look at

and fix for each test.

#41 - 06/18/2018 05:50 PM - pespin

- Related to Bug #3353: osmo-bts: TTCN3 BTS_Tests.TC_rll_est_ind fail behavior not clear added

#42 - 06/18/2018 06:04 PM - pespin

Here's a full log of the "Dynamic test case error", it already happened during the first test of the full suite, so it doesn't seem to be related to state left from previous tests:

```
MC2> smtc
Executing all items of [EXECUTE] section.
MC2> MTC@f58d536352bc: Starting external command `../ttcn3-tcpdump-start.sh BTS_Tests.TC_rll_est_ind'.
Waiting for tcpdump to start... 0
MTC@f58d536352bc: External command `../ttcn3-tcpdump-start.sh BTS_Tests.TC_rll_est_ind' was executed successfu
lly (exit status: 0).
MTC@f58d536352bc: Test case TC_rll_est_ind started.
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: IPA: Connected
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: CCM Tx:{ msg_type := IPAC_MSGT_ID_GET (4), u := { get := { { len := 1,
 tag := IPAC_IDTAG_UNITNAME (1) } } } }
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: CCM Rx:{ msg_type := IPAC_MSGT_ID_RESP (5), u := { resp := { { len :=
28, tag := IPAC_IDTAG_UNITNAME (1), data := "sysmoBTS-00-00-00-00-00-00" & char(0, 0, 0, 0) } } } }
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: IPA ID RESP: { { len := 28, tag := IPAC_IDTAG_UNITNAME (1), data := "s
ysmoBTS-00-00-00-00-00-00" & char(0, 0, 0, 0) } }
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: CCM Tx:{ msg_type := IPAC_MSGT_ID_ACK (6), u := omit }
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: CCM Rx:{ msg_type := IPAC_MSGT_ID_ACK (6), u := omit }
MTC@f58d536352bc: Setting RSL_SYSTEM_INFO_3 (3): { header := { l2_plen := { l2_plen := 18, zero_one := '01'B }
, skip_indicator := 0, rr_protocol_discriminator := 6, message_type := SYSTEM_INFORMATION_TYPE_3 (27) }, paylo
ad := { si3 := { cell_id := 23, lai := { mcc_mnc := '262F42'H, lac := 42 }, ctrl_chan_desc := { msc_r99 := tru
e, att := true, bs_ag_blks_res := 1, ccch_conf := CCHAN_DESC_1CCCH_COMBINED (1), si22ind := false, cbq3 := CBQ
3_IU_MODE_NOT_SUPPORTED (0), spare := '00'B, bs_pa_mfrms := 0, t3212 := 1 }, cell_options := { dn_ind := false
, pwrc := false, dtx := MS_MAY_USE_UL_DTX (0), radio_link_tout_div4 := 7 }, cell_sel_par := { cell_resel_hyst_
2dB := 2, ms_txpwr_max_cch := 7, acs := '0'B, neci := true, rxlev_access_min := 0 }, rach_control := { max_ret
rans := RACH_MAX_RETRANS_7 (3), tx_integer := '1001'B, cell_barr_access := false, re_not_allowed := true, acc
 := '0000010000000000'B }, rest_octets := '2B2B2B2B'O ("++++") } } }
MTC@f58d536352bc: Setting RSL_SYSTEM_INFO_3 (3): '49061B001762F224002AC90001074740E504002B2B2B2B'O
MTC@f58d536352bc: Setting RSL_SYSTEM_INFO_2 (2): { header := { l2_plen := { l2_plen := 22, zero_one := '01'B }
, skip_indicator := 0, rr_protocol_discriminator := 6, message_type := SYSTEM_INFORMATION_TYPE_2 (26) }, paylo
ad := { si2 := { bcch_freq_list := '00000000000000000000000000000000'O, ncc_permitted := '11111111'B, rach_con
trol := { max_retrans := RACH_MAX_RETRANS_7 (3), tx_integer := '1001'B, cell_barr_access := false, re_not_allo
wed := true, acc := '0000010000000000'B } } } }
MTC@f58d536352bc: Setting RSL_SYSTEM_INFO_2 (2): '59061A00000000000000000000000000000000FFE50400'O
MTC@f58d536352bc: Setting RSL_SYSTEM_INFO_4 (4): { header := { l2_plen := { l2_plen := 12, zero_one := '01'B }
, skip_indicator := 0, rr_protocol_discriminator := 6, message_type := SYSTEM_INFORMATION_TYPE_4 (28) }, paylo
ad := { si4 := { lai := { mcc_mnc := '262F42'H, lac := 42 }, cell_sel_par := { cell_resel_hyst_2dB := 2, ms_tx
pwr_max_cch := 7, acs := '0'B, neci := true, rxlev_access_min := 0 }, rach_control := { max_retrans := RACH_MA
X_RETRANS_7 (3), tx_integer := '1001'B, cell_barr_access := false, re_not_allowed := true, acc := '00000100000
00000'B }, cbch_chan_desc := omit, cbch_mobile_alloc := omit, rest_octets := ''O } } }
MTC@f58d536352bc: Setting RSL_SYSTEM_INFO_4 (4): '31061C62F224002A4740E50400'O
MTC@f58d536352bc: "TC_rll_est_ind": XXX Starting { sapi := 0, link_id := { c := FACCH_SDCCH (0), na := false,
prio := SAPI0_PRIO_NORMAL (0), sapi := 0 }, l3 := '01020304'O, exp := true } on { u := { sdcch4 := { tag := '0
01'B, sub_chan := 2 } }, tn := 0 }
MTC@f58d536352bc: "TC_rll_est_ind": XXX Starting { sapi := 0, link_id := { c := FACCH_SDCCH (0), na := false,
prio := SAPI0_PRIO_NORMAL (0), sapi := 0 }, l3 := ''O, exp := true } on { u := { sdcch4 := { tag := '001'B, su
b_chan := 2 } }, tn := 0 }
TC_rll_est_ind-RSL(4)@f58d536352bc: Dynamic test case error: Data cannot be sent on port CLIENT_PT to componen
t 6 because there is no connection towards component 6.
TC_rll_est_ind-RSL-IPA(3)@f58d536352bc: Dynamic test case error: Port IPA_RSL_PORT has neither connections nor
 mappings. Message cannot be sent on it.
MTC@f58d536352bc: "TC_rll_est_ind": XXX Starting { sapi := 3, link_id := { c := FACCH_SDCCH (0), na := false,
prio := SAPI0_PRIO_NORMAL (0), sapi := 3 }, l3 := '01020304'O, exp := false } on { u := { sdcch4 := { tag := '
001'B, sub_chan := 2 } }, tn := 0 }
MTC@f58d536352bc: Dynamic test case error: Error message was received from MC: The connect operation refers to
 test component with component reference 4, which has already terminated.
MTC@f58d536352bc: Test case TC_rll_est_ind finished. Verdict: error
MTC@f58d536352bc: Starting external command `../ttcn3-tcpdump-stop.sh BTS_Tests.TC_rll_est_ind error'.
Waiting for tcpdump to finish... 0 (prev_count=-1, count=8815)
Waiting for tcpdump to finish... 1 (prev_count=8815, count=9550)
MTC@f58d536352bc: External command `../ttcn3-tcpdump-stop.sh BTS_Tests.TC_rll_est_ind error' was executed succ
essfully (exit status: 0).
MC@f58d536352bc: Test execution finished.
```

However, I have been running this test several times alone, and now that I switched back to running BTS_Tests.control again, it strikes.

#43 - 07/18/2018 09:46 AM - pespin

- Status changed from In Progress to Stalled

Waiting until patches improving error detection / stopping tests at the right time are merged before continuing with looking at issues appearing in different tests, it will make my life easier:
https://gerrit.osmocom.org/#/c/osmo-ttcn3-hacks/+/9905/
https://gerrit.osmocom.org/#/c/osmo-ttcn3-hacks/+/9907/

Once they are merged, I need to generate a new docker file for TTCN3 tests and run the tests.

#44 - 09/18/2018 03:58 PM - pespin

I had a look at current status of the test setup today and I found out most tests failed while trying to send a RESET.conf message through L1CTL to osmocon. Further investigation showed that osmocon starts and enters an infinite loop loading the firmware to RAM forever.

I checked several versions to make sure it was not a software regression, and it's not. I tested with an old sysroot (osmocon bin + libosmocore + osmocombb fw) that is known to have worked, and I saw same issue. Exact same binaries work fine in RnD setup.

Power-cycling the prod setup didn't help. It looks like a physicial or device related issue, I asked roh for support regarding the physisical HW aspects and he should look at it soon.

#45 - 09/25/2018 11:41 AM - pespin

I'm running the tests now in osmo-gsm-tester RnD setup since there are issues with the motorola+osmocon in prod setup (infinite loop loading fw).

These are more or less the test set failing:

```
BTS_Tests.TC_meas_res_sign_tchf fail
BTS_Tests.TC_meas_res_sign_tchh_toa256 fail
BTS_Tests.TC_paging_tmsi_200percent fail
BTS_Tests.TC_si_sched_13_2bis_2ter_2quater fail
BTS_Tests.TC_pcu_rach_content fail
BTS_Tests.TC_rll_est_ind fail
BTS_Tests.TC_rll_rel_ind_DCCH_0 fail
BTS_Tests.TC_rll_rel_ind_DCCH_3 fail
BTS_Tests.TC_rll_rel_ind_ACCH_0 fail
BTS_Tests.TC_rll_rel_ind_ACCH_3 fail
BTS_Tests.TC_rll_rel_req fail
BTS_Tests.TC_encr_cmd_a51 fail
```

I started looking at "TC_meas_res_sign_tchf", which fails this way:

```
10:44:55.614221 131 BTS_Tests.ttcn:1299 setverdict(fail): pass -> fail reason: "Received unspecific MEAS RES {
 msg_disc := { msg_group := RSL_MDISC_DCHAN (4), transparent := false }, msg_type := RSL_MT_MEAS_RES (40), ies
 := { { iei := RSL_IE_CHAN_NR (1), body := { chan_nr := { u := { ch0 := RSL_CHAN_NR_Bm_ACCH (1) }, tn := 1 } }
 }, { iei := RSL_IE_MEAS_RES_NR (27), body := { meas_res_nr := 1 } }, { iei := RSL_IE_UPLINK_MEAS (25), body :
= { uplink_meas := { len := 3, rfu := '0'B, dtx_d := false, rxlev_f_u := 53, reserved1 := '00'B, rxlev_s_u :=
53, reserved2 := '00'B, rxq_f_u := 0, rxq_s_u := 0, supp_meas_info := omit } } }, { iei := RSL_IE_BS_POWER (4)
, body := { bs_power := { reserved := 0, epc := false, fpc := false, power_level := 0 } } }, { iei := RSL_IE_L
1_INFO (10), body := { l1_info := { ms_power_lvl := 7, fpc := false, reserved := 0, actual_ta := 0 } } }, { ie
i := RSL_IE_L3_INFO (11), body := { l3_info := { len := 18, payload := '0615282801C00000000000000000000000000'O
 } } }, { iei := RSL_IE_MS_TIMING_OFFSET (37), body := { ms_timing_offset := 63 } } } }", new component reason
: "Received unspecific MEAS RES { msg_disc := { msg_group := RSL_MDISC_DCHAN (4), transparent := false }, msg_
type := RSL_MT_MEAS_RES (40), ies := { { iei := RSL_IE_CHAN_NR (1), body := { chan_nr := { u := { ch0 := RSL_C
HAN_NR_Bm_ACCH (1) }, tn := 1 } } }, { iei := RSL_IE_MEAS_RES_NR (27), body := { meas_res_nr := 1 } }, { iei :
= RSL_IE_UPLINK_MEAS (25), body := { uplink_meas := { len := 3, rfu := '0'B, dtx_d := false, rxlev_f_u := 53,
reserved1 := '00'B, rxlev_s_u := 53, reserved2 := '00'B, rxq_f_u := 0, rxq_s_u := 0, supp_meas_info := omit }
} }, { iei := RSL_IE_BS_POWER (4), body := { bs_power := { reserved := 0, epc := false, fpc := false, power_le
vel := 0 } } }, { iei := RSL_IE_L1_INFO (10), body := { l1_info := { ms_power_lvl := 7, fpc := false, reserved
 := 0, actual_ta := 0 } } }, { iei := RSL_IE_L3_INFO (11), body := { l3_info := { len := 18, payload := '06152
82801C00000000000000000000000000'O } } }, { iei := RSL_IE_MS_TIMING_OFFSET (37), body := { ms_timing_offset :=
63 } } } }"
```

So it fails to match the specific measurements created at "f_TC_meas_res_periodic". It seems related to the fact that we don't use mp_bb_trxc_port in the HW setup because we don't use trxcon there (mp_bb_trxc_port = -1):

```
    if (mp_bb_trxc_port != -1) {
        g_pars.l1_pars.meas_ul.full.rxlev := dbm2rxlev(-100);
        f_trxc_fake_rssi(100);

        g_pars.l1_pars.timing_offset_256syms := 512; /* 2 symbols */
        f_trx_fake_toffs256(g_pars.l1_pars.timing_offset_256syms);
    } else {
        g_pars.l1_pars.timing_offset_256syms := 0; /* FIXME */
        g_pars.l1_pars.meas_ul.full.rxlev := dbm2rxlev(-55); /* FIXME */
    }
```

So those FIXME should be fixed, the question is what's actually th best procedure to do so. I'm sure "TC_meas_res_sign_tchh_toa256" fails due to same reason.

#46 - 09/25/2018 12:20 PM - laforge

On Tue, Sep 25, 2018 at 11:41:34AM +0000, pespin [REDMINE] wrote:

> So those FIXME should be fixed, the question is what's actually th best procedure to do so.

- run the test in the given setup
- record the min/max/avg values for timing and rxlev
- enter them as a range of permitted values in the receive templates in BTS_Tests.ttcn

I you want to make it less hard-coded, one could have the values exposed as module
paraemters in to the config file. I think it's best to enter/specify them is the following way:

- expected (average/mean) value
- permitted tolerance (in percent, or absolute value)

Then the template would permit the range (mean - tolerance) .. (mean + tolerance)
as the permitted range.

With some luck, the setup-specific value that needs to change is the expected value,
as this depends on the attenuation between transmitter and receiver.
The tolerance should be the same on every setup/installation, as this is only determined
by the overall tolerances of the various elements used.

#47 - 09/25/2018 01:07 PM - fixeria

It should be noted that the (RXLEV) measurements produced by a phone running OsmocomBB firmware
may be inaccurate, because reading and applying the unit calibration data is not supported yet (see #3582).

#48 - 09/25/2018 03:31 PM - pespin

- Related to Bug #3025: BTS_Tests.TC_paging_tmsi_200percent fails added

#49 - 09/27/2018 10:36 AM - pespin

- File pcap-recorder_any(filters=_host 10.42.42.8 and port not 22_).pcap added

Regarding discussion in https://gerrit.osmocom.org/#/c/osmo-ttcn3-hacks/+/11083/2/bts/BTS_Tests.ttcn@58 with test "BTS_Tests.TC_meas_res_sign_tchf" having the motorola CXX phone sending different values for ms_power_level (0 and requested 7).

I checked the pcap file of a complete test run and it seems both values are more a les sdistributed over time, that is, we don't get a few ms_power_level=0 at start then ms_power_level=7 starting from some point.

I attach the GSMTAP pcap file of the test to show it. You can find the ms_power_level values by filtering Measurement Report (gsm_a.dtap.msg_rr_type == 0x15) and then looking at "SACCH L1 Header".

Some details of what I see:

- It seems most Measurement Report messages come with ms_power_level=0 (59msgs vs 8msgs).
- Usually Measurement Report messages with ms_power_level=7 come in pairs (2 consecutive).
- Measurement Report messages with ms_power_level=7 contain:
  - L1 Header FPC=1: In use
  - RXLEV: -71 <= x < -70 dBm (40)
  - NO-NCELL-M: Neighbour cell information not available for serving cell (7)
- On the other hand, Measurement Report messages with ms_power_level=0 contain:
  - L1 Header FPC=0: Not in use
  - RXLEV:-88 <= x < -87 dBm (23)
  - NO-NCELL-M: No neighbour cell measurement result (0)

#50 - 09/27/2018 11:10 AM - laforge

On Thu, Sep 27, 2018 at 10:36:27AM +0000, pespin [REDMINE] wrote:

> Some details of what I see:
>
> - It seems most Measurement Report messages come with ms_power_level=0 (59msgs vs 8msgs).
> - Usually Measurement Report messages with ms_power_level=7 come in pairs (2 consecutive).
> - Measurement Report messages with ms_power_level=7 contain:
>   - L1 Header FPC=1: In use

this is clearly wrong. We don't do fast power control (FPC) in OsmoBTS, but only normal power control.

#51 - 09/27/2018 02:16 PM - pespin

It's strange because in those messages, the FPC field is taken from 1 bit which is also part of the ms_power_level field value, so I'm pretty sure it doesn't apply here or somehow wireshark is wrong. Otherwise it makes no sense to me.

I have been investigating how it comes the motorola phone is sometimes sending ms_power_level 7 and sometimes 0, by looking at osmocombb and L1CTL TTCN parts. In osmocombb, the interesting related variable is called "l1s.tx_power" everywhere. It seems this value is set to default value of 7 when the state is reset (l1s_reset(), l1ctl_rx_reset_req(), L1CTL_RESET_REQ). Other than that reset code path, l1s.tx_power can only be set to another value by using l1ctl_rx_param_req() which can only be called by sending an L1CTL message to it: L1CTL_PARAM_REQ. In TTCN3, that message is only sent by template ts_L1CTL_PAR_REQ, which is only used in function f_L1CTL_PARAM(), which is never used in any TTCN3 code.

So as a result, we have a received tx_power changing but nobody seems to ever be telling the motorola CXX phone to change it.

I then did the following changes:

- In TTCN3, Set ms_txpwr_max_cch=9 (to be different than the default 7 in osmocombb fw code).
- Do following change, to force tx_power to some value (different than the default 7 in osmocombb fw code, and different than the one received by network).

```
@@ -1363,7 +1363,10 @@ private function f_est_dchan(boolean encr_enable := false) runs on ConnHdlr {
        var GsmFrameNumber fn;
        var ImmediateAssignment imm_ass;
        var integer ra := 23;
+       var integer ta := 0;
```

```
+       var integer tx_power := 8;

+       f_L1CTL_PARAM(L1CTL, ta, tx_power);
        fn := f_L1CTL_RACH(L1CTL, ra);
```

Result: I get same pattern as before, but with 0 and 8. Which means none of the default value nor the network values are being used, and still 0 is sometimes sent which apparently comes from nowhere.

#52 - 09/27/2018 06:10 PM - pespin

- Status changed from Stalled to In Progress

Submitted patch for FPC issue seen in wireshark: https://code.wireshark.org/review/#/c/29893/

Regarding the different values in ms_power_level, it seems related to f_L1CTL_PARAM only controlling values for dummy meas report which are sent if no meas report message is received from upper layer on time.
The upper layer messages are crafted in TTCN3 function as_l1_sacch(), and the 2 octets or SACCH L1 HEADER are hardcoded to 0: "var octetstring pl := '0000'O & enc_LapdmFrameAB(lb);"

#53 - 09/27/2018 07:07 PM - fixeria

> Regarding the different values in ms_power_level, it seems related to f_L1CTL_PARAM only controlling values for dummy meas report which are sent if no meas report message is received from upper layer on time.

Actually, this is the problem I was talking about in:

https://lists.osmocom.org/pipermail/baseband-devel/2018-March/005513.html

This dummy measurement report shall not "expose" the actual TA and TX power level values
(i.e. which are used by the hardware) as it breaks the idea of distance spoofing...

Instead of putting the actual values into a static template, it makes sense to cache
the last received measurement report and use it until the next one arrives.

But, in any case this is a problem of OsmocomBB, and weakly related to the issue topic.
I just need to find some time and finally fix this in both firmware and trxcon.

> The upper layer messages are crafted in TTCN3 function as_l1_sacch(), and the 2 octets or SACCH L1 HEADER are hardcoded to 0: "var octetstring pl := '0000'O & enc_LapdmFrameAB(lb);"

I think this is exactly the reason of Measurement Reports with different values you observed.

There are several possible solutions:

1) The simplest solution is to call f_L1CTL_PARAM(ta = 0, tx_power = 0), which isn't called anywhere yet.
This would make all Measurement Reports contain the same values. Not sure if this is the best solution.

Please note that tx_power is being set to default value by 'layer1/sync.c/l1s_reset()', so f_L1CTL_PARAM() should be called **after** reset is done.

2) Fix the problem I mentioned above, so all Measurement Reports would always contain the indicated parameters (tx_power=0 and ta=0), despite the actual values used by HW/FW would remain default.

I also think it makes sense to make the L1 SACCH header configurable in BTS_Tests.ttcn/as_l1_sacch()...

#54 - 10/04/2018 03:57 PM - roh

i think i found the issue with the c118 not being reliable/loading in circles:

the cable-ties mounting it to the testbed were too tight and there was no backing behind the phone (while using glued-in-place rf cable).
this resuled in the case torqueing a bit, which was enough for the battery simulator board making bad contact via the battery pins.
such the phone tried to load, but failed in the process due to brown-out.

i changed the board (cutting the leads protruding short as possible) and made the contact pads thicker by adding some metallic pieces. i also changed the mechanics to hold the phone in a more straight position and added some padding material below before mounting it with zip-ties again.

it re-enumerated as ttyUSB1 /dev/serial/by-path/pci-0000:00:12.2-usb-0:5.4.3:1.0-port0

please test

#55 - 10/09/2018 04:34 PM - pespin

After HW fix, motorola c118 phone is working fine again in the prod setup.

I took over looking at failing tests in https://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_ttcn3/test_results_analyzer/

I had a look at **TC_encr_cmd_a51**, which seems to be failing most of the time but it recently passed twice in a row (#650 and #649). I see a similar pattern for **TC_encr_cmd_a52**.

In the failing case (run #651 and #646):

```
09:43:01.047017 450 BTS_Tests.ttcn:3793 Timeout T: 3 s
09:43:01.047367 450 BTS_Tests.ttcn:3794 setverdict(fail): pass -> fail reason: "Timeout waiting for DATA_IND",
 new component reason: "Timeout waiting for DATA_IND"
```

Looking at log files and pcap file, the failure is triggered because according to BTS GSMTAP the MS apparently is not sending the first ciphered I-frame after receiving RSL ENC CMD (with Ciphering Mode Command inside):

```
/* Test unencrypted channel activation followed by explicit ENCR CMD later */
function f_TC_encr_cmd(charstring id) runs on ConnHdlr {
...
    /* then send the RSL ENCR CMD with an actual RR CIPH MOD CMD inside */
    RSL.send(ts_RSL_ENCR_CMD(g_chan_nr, link_id, g_pars.encr.alg_id, g_pars.encr.key, l3));
    /* expect the L3 to arrive still unencrypted on the MS side */
    f_l1_exp_lapdm(tr_LAPDm_I(link_id.sapi, cr_MT_CMD, ?, ?, ?, l3));                          <--------- THIS
 IS RECEIVED FINE
```

```
    /* configure L1 to apply ciphering */
    var uint8_t alg_id := f_alg_id_to_l1ctl(g_pars.encr.alg_id);
    f_L1CTL_CRYPTO_REQ(L1CTL, g_pars.chan_nr, alg_id, g_pars.encr.key);

    /* send first ciphered I-frame in response and expect it on RSL */
    f_data_mo(link_id, true, 1, 0, '0a0b0c0d'O, exp_sacch := true);                          <--------- THIS N
EVER APPEARS IN BTS GSTMAP
```

Test fails actually after second round (it's run multiple times for different channels), while using { u := { lm := { tag := '0001'B, sub_chan := 1 } }, tn := 5 } and as shown in wireshark: "Lm + ACCH (sub-chan 1) (3)". The first round "Bm + ACCH (1)" is fine.

If we look at other failing runs, like #648 and #647, both fail in a similar way, while sending an uplink message during 2nd round (Lm+ACCH) which never shows up in GSMTAP, but does not fail exactly at the same place. In this case, both fail during establishing ABM after dedicated channel has been established:
"f_TC_encr_cmd --> f_est_rll_mo(link_id.sapi, link_id, '23420815'O); --> f_tx_lapdm(ts_LAPDm_SABM(sapi, cr_MO_CMD, true, l3), link_id);"

So it seems osmocom-bb "randomly" fails to send data correctly over TCH/H+ACCH, or osmo-bts-trx/osmo-trx fail to receive it. That's why sometimes the test fails during one step, sometimes during next step, and sometimes passes.

#56 - 10/10/2018 10:43 AM - pespin

Attaching archive with osmo-gsm-tester+ttcn3 run information during TC_chan_act_a51() run, which has exactly same issue as TC_encr_cmd_*() tests explained above. Inside the archive there's both a successful run and a failing run, to be able to check differences.

#57 - 10/10/2018 10:43 AM - pespin

- File chan_act_a51-failpass.tar.gz added

#58 - 10/10/2018 03:18 PM - fixeria

Hi Pau,

> So it seems osmocom-bb "randomly" fails to send data correctly over TCH/H+ACCH,
> or osmo-bts-trx/osmo-trx fail to receive it. That's why sometimes the test fails
> during one step, sometimes during next step, and sometimes passes.

I guess this is a problem of the OsmocomBB firmware. FACCH/H can be initiated at specific frame numbers only, so ignoring this rule may result in misaligned FACCH/H transmission.

#59 - 10/11/2018 03:00 PM - pespin

Checking at FACCH/H transmission scheduling the used FNs seems fine. According to the code, the DSP expects to receive the message on the preceding burst, and that's what it does correctly.
I also tried switching to expected and not preceding bursts just in case, but got same results during TTCN3 tests, that is, sometimes fails and sometimes pass.

```
--- a/src/target/firmware/layer1/prim_tch.c
+++ b/src/target/firmware/layer1/prim_tch.c
@@ -402,17 +402,15 @@ static int l1s_tch_cmd(__unused uint8_t p1, __unused uint8_t p2, uint16_t p3)
                icnt++;

        /* Load FACCH data if we start a new burst */
-       /* (the DSP wants the data on the CMD of the burst _preceding_ the
-        * first burst) */
        if (tch_f_hn) {
                /* FACCH/F: B0(0...7),B1(4...11),B2(8...11,0...3) */
                facch_tx_now = ((l1s.next_time.fn % 13) % 4) == 3;
        } else {
                /* FAACH/H: See GSM 05.02 Clause 7 Table 1of9 */
                uint8_t t2_norm = l1s.next_time.t2 - tch_sub;
-               facch_tx_now = (t2_norm == 23) ||
-                              (t2_norm ==  6) ||
-                              (t2_norm == 15);
+               facch_tx_now = (t2_norm ==  0) ||
+                              (t2_norm ==  8) ||
+                              (t2_norm == 17);
        }
```

As requested by fixeria, I also tested adding a sleep after sending L1CTL_CRYPTO_REQ to make sure the DSP is not applying encryptin in the middle of the burst block, but I got same results:

```
--- a/bts/BTS_Tests.ttcn
+++ b/bts/BTS_Tests.ttcn
@@ -1399,6 +1399,7 @@ private function f_est_dchan(boolean encr_enable := false) runs on ConnHdlr {
        if (encr_enable) {
                var uint8_t alg_id := f_alg_id_to_l1ctl(g_pars.encr.alg_id);
                f_L1CTL_CRYPTO_REQ(L1CTL, g_pars.chan_nr, alg_id, g_pars.encr.key);
+               f_sleep(2.0); /* make sure encryption is applied */
        }

        g_first_meas_res := true;
@@ -3863,7 +3864,7 @@ function f_TC_encr_cmd(charstring id) runs on ConnHdlr {
        /* configure L1 to apply ciphering */
        var uint8_t alg_id := f_alg_id_to_l1ctl(g_pars.encr.alg_id);
        f_L1CTL_CRYPTO_REQ(L1CTL, g_pars.chan_nr, alg_id, g_pars.encr.key);
-
+       f_sleep(2.0); /* make sure encryption is applied */
        /* send first ciphered I-frame in response and expect it on RSL */
        f_data_mo(link_id, true, 1, 0, '0a0b0c0d'O, exp_sacch := true);
```

#60 - 10/20/2018 07:29 PM - laforge

Adding tnt here to the list of followers. The plan is that he can help with investigating any test failures of tests that pass in the virtual environment but fail in the real hardware environment. He has extensive Layer0/Layer1 knowledge and should be able to help where pespin is a bit out of his usual comfort zone.

The problems could likely be in the test case expectations, in OsmocomBB or in OsmoBTS.

It may make sense to start more tickets about individual issues/failures rather than adding more to this generic ticket.

Files

cp210x-dkms_4.16-0.tar.gz	13.9 KB	05/26/2018	laforge
cp210x-dkms_4.16-0.dsc	507 Bytes	05/26/2018	laforge
cp210x-4.16.tar.bz2	13.6 KB	05/26/2018	laforge
cp210x-dkms_4.16-0_all.deb	13.8 KB	05/26/2018	laforge
pcap-recorder_any(filters=_host 10.42.42.8 and port not 22_).pcap	418 KB	09/27/2018	pespin
chan_act_a51-failpass.tar.gz	363 KB	10/10/2018	pespin