

## OsmoBTS - Bug #2325

### sporadic shutdown of osmo-bts-trx in osmo-gsm-tester runs (no clock from trx)

06/13/2017 02:52 PM - neels

<b>Status:</b>	Stalled	<b>Start date:</b>	06/13/2017
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>	pespin	<b>% Done:</b>	20%
<b>Category:</b>	osmo-bts-trx		
<b>Target version:</b>	osmo-bts-trx refresh		
<b>Spec Reference:</b>			

#### Description

The osmo-bts-trx process sometimes dies prematurely during osmo-gsm-tester runs using the Ettus B210. The cause is not clear yet, no core file seems to be created.

For example, see [http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester\\_run/591/](http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_run/591/)

#### Related issues:

Related to OsmoGSMTester - Feature #2327: document NTP as cause for failing o...	<b>Rejected</b>	<b>06/14/2017</b>
Related to OsmoBTS - Bug #2339: osmo-bts-trx uses non-monotonic system clock ...	<b>Closed</b>	<b>06/24/2017</b>
Related to OsmoTRX - Bug #2344: OsmoTRX is not using SCHED_RR	<b>Stalled</b>	<b>06/29/2017</b>

#### History

##### #1 - 06/13/2017 10:46 PM - neels

This happens quite frequently, apparently more than one out of ten runs

##### #2 - 06/13/2017 10:54 PM - neels

instances of this are seen in abovementioned run 591, again in 595, 597, 599.

Each run has two trx runs, so, roughly: out of 20 runs, four hit the "Process ended prematurely" for osmo-bts-trx.

##### #3 - 06/13/2017 10:55 PM - neels

- Related to Bug #2321: osmo-gsm-tester: store properly coredump files when a process crashes added

##### #4 - 06/14/2017 12:41 AM - neels

- Related to deleted (Bug #2321: osmo-gsm-tester: store properly coredump files when a process crashes)

##### #5 - 06/14/2017 12:50 AM - neels

The sporadic "crashes" are actually intentional shutdown by osmo-bts-trx:

```
20170614021016906 DOML <0001> bts.c:208 Shutting down BTS 0, Reason No clock from osmo-trx
```

The end of the log shows "Shutdown timer expired", which has a three second expiry time. About 30 logging lines above that, the shutdown reason is logged.

The "No clock" is logged about two seconds after the OML has successfully set up the BTS. Immediately following that, the OML is torn down again, concluding in

```
Shutdown timer expired
```

I am not so sure about how to fix this, will have to ask osmo-trx guys.

**#6 - 06/14/2017 01:35 AM - neels**

- Status changed from New to In Progress
- Assignee set to neels

<https://gerrit.osmocom.org/2909> <-- sets l1c logging level to notice

results:

```
20170614032014399 DRSL <0000> rsl.c:2333 (bts=0,trx=0,ts=0,ss=2) Fwd RLL msg EST_IND from LAPDm to A-bis
20170614032018533 DL1C <0006> scheduler_trx.c:1451 PC clock skew: elapsed uS 4136730
20170614032018533 DOML <0001> bts.c:208 Shutting down BTS 0, Reason No clock from osmo-trx
20170614032018533 DL1C <0006> scheduler.c:240 Exit scheduler for trx=0
20170614032018533 DL1C <0006> scheduler.c:216 Init scheduler for trx=0
20170614032018533 DOML <0001> oml.c:280 OC=RADIO-CARRIER INST=(00,00,ff) AVAIL STATE OK -> Off line
[...]
Shutdown timer expired
```

I asked on openbsc@: <http://lists.osmocom.org/pipermail/openbsc/2017-June/010802.html>

**#7 - 06/14/2017 03:44 PM - neels**

- Related to Feature #2327: document NTP as cause for failing osmo-bts-trx runs in osmo-gsm-manuals? added

**#8 - 06/14/2017 03:45 PM - neels**

Vadim indicated NTP as possible cause for clock skews. I switched off the ntp service on the main unit now. It will come back upon reboot, so we may need to switch it off again. If this proves to be the cause of failures, we can uninstall ntp completely or something.

**#9 - 06/24/2017 12:12 AM - laforge**

- Category set to osmo-bts-trx

FYI:

- osmo-bts uses gettimeofday() to determine how much system time has expired between two FN indications from OSmoTRX
- the maximum permitted value is  $50 \times 4.6\text{ms}$ , i.e. 50 TDMA frames, totalling to 230750, i.e. 230.75ms
- the actual skew as reported here is 4.13 seconds

I assume that the -r or --realtime argument is used when starting osmo-bts-trx, at least [http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester\\_run/591/](http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_run/591/) indicates so.

**#10 - 06/24/2017 11:01 AM - laforge**

- Related to Bug #2339: osmo-bts-trx uses non-monotonic system clock for frame number timer added

**#11 - 06/24/2017 11:09 AM - laforge**

**#12 - 06/28/2017 11:52 PM - laforge**

- Assignee changed from needs to pespín

- % Done changed from 0 to 20

I've just tried to reproduce this on my laptop (no APU board here at home), but couldn't.

I've used 'stress-ng' with various options, such as stress-ng --vm 4 --exec 4 --hdd 4 --netdev 4 --sync-file 4 to generate a combination of i/o and cpu load.

osmo-bts-trx was running with real-time priority and was completely unaffected, even at a system load > 30.

When running without real-time priority, I got the occasional

```
<0006> scheduler_trx.c:1587 FN timer expire_count=3: We missed 2 timers
```

but that's not 44 frame periods, and osmo-bts-trx continues to live.

Let's try that on an APU board next.

**#13 - 06/29/2017 12:43 PM - laforge**

Tried to reproduce this on an PCEngines APU1D4 with Debian 8. No success.

Even while stressing the system using stress-ng --vm 4 --hdd 4 --cpu 4 --open 4 --sem 4 --sock 4 --float 2 --io 2 --timer 2 while running osmo-bts-trx and osmo-bts, I was not able to get more than the occasional

```
<0006> scheduler_trx.c:1718 We were 1 FN slower than TRX, compensated  
<0006> scheduler_trx.c:1704 We were 1 FN faster than TRX, compensating  
<0006> scheduler_trx.c:1718 We were 2 FN slower than TRX, compensated  
<0006> scheduler_trx.c:1571 FN timer expire_count=2: We missed 1 timers
```

which is all expected. So whatever it is, it must be something specific to the osmo-gsm-tester. Maybe it's enabling excessive logging to a file or stderr, and then that logging is blocking the process?

**#14 - 06/29/2017 12:44 PM - laforge**

During startup, I get a couple of

```
<0006> scheduler_trx.c:1704 We were 39 FN faster than TRX, compensating
```

but that's before all initialization is done, i.e. before the system is really running.

**#15 - 06/29/2017 12:53 PM - laforge**

[http://git.osmocom.org/osmo-gsm-tester/tree/src/osmo\\_gsm\\_tester/templates/osmo-bts-trx.cfg.tmp](http://git.osmocom.org/osmo-gsm-tester/tree/src/osmo_gsm_tester/templates/osmo-bts-trx.cfg.tmp) doesn't look like there is any file-based logging, nor that there's lots of logging enabled to begin with.

**#16 - 06/29/2017 01:07 PM - laforge**

- Related to Bug #2344: OsmoTRX is not using SCHED\_RR added

**#17 - 06/29/2017 01:55 PM - pespin**

We do redirect stderr and stdout to files, as can be seen here:

[http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester\\_run/ws/trial-1036/last\\_run/sms:trx/mo\\_mt\\_sms.py/osmo-bts-trx/osmo-bts-trx/stderr/\\*view\\*/](http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_run/ws/trial-1036/last_run/sms:trx/mo_mt_sms.py/osmo-bts-trx/osmo-bts-trx/stderr/*view*/)

However the amount of data written doesn't seem to be super big. I'll try anyway to filter the output a lot more and run some tests to be sure that's not the issue.

**#18 - 07/03/2017 11:23 AM - pespin**

Disabling logging to stderr doesn't fix the issue.

On the other hand, I detected that I'm not able to reproduce the issue unless I power up one modem to connect to the BTS.

**#19 - 07/03/2017 05:50 PM - pespin**

I have seen at least once osmo-bts-trx stopping with a failure printing that 65 timerfd events had expired before we read them. Each event is triggered around 120/26ms, which means then it took us more than 300ms to read the FD after it was first signaled. The only option I can think of is that the system is really congested or that this process is starving.

I checked the priorities on the osmo-gsm-tester main unit while running the tests:

```
# for p in $(ls /proc/$(pidof osmo-trx)/task/); do chrt -p $p; done
pid 4214's current scheduling policy: SCHED_OTHER
pid 4214's current scheduling priority: 0
pid 4236's current scheduling policy: SCHED_OTHER
pid 4236's current scheduling priority: 0
pid 4237's current scheduling policy: SCHED_OTHER
pid 4237's current scheduling priority: 0
pid 4257's current scheduling policy: SCHED_OTHER
pid 4257's current scheduling priority: 0
```

```

pid 4287's current scheduling policy: SCHED_OTHER
pid 4287's current scheduling priority: 0
pid 4337's current scheduling policy: SCHED_RR
pid 4337's current scheduling priority: 43
pid 4338's current scheduling policy: SCHED_RR
pid 4338's current scheduling priority: 44
pid 4339's current scheduling policy: SCHED_RR
pid 4339's current scheduling priority: 45
pid 4340's current scheduling policy: SCHED_RR]
pid 4340's current scheduling priority: 42
pid 4341's current scheduling policy: SCHED_RR
pid 4341's current scheduling priority: 40

# for p in $(ls /proc/$(pidof osmo-bts-trx)/task/); do chrt -p $p; done
pid 4570's current scheduling policy: SCHED_RR
pid 4570's current scheduling priority: 1

```

As reading from <http://man7.org/linux/man-pages/man7/sched.7.html> , that means that the priorities of osmo-trx threads are a lot higher than the one used in osmo-bts-trx, and I see those thread taking a lot of CPU, so it could indeed be happening that the osmo-bts-trx process is starving (it tries to keep up due to RR but even then not good enough).

I'll try to do some tests:

- Increase osmo-bts-trx rt priority to 50 or similar.
- Pin osmo-bts-trx to one core and all threads from osmo-trx to another core.

**#20 - 07/04/2017 12:06 PM - pespin**

Setting SCHED\_RR with priority 50 on osmo-bts-trx (-r 50) didn't help to improve the issue.  
Pinning osmo-trx to one core and osmo-bts-trx to the other one didn't help neither. (taskset -a -c \$cpunum \$binary)

The issue seems to be during the initial seconds when both osmo-bts-trx and osmo-trx are powered on and start the configuration phase. Once the PH-RTS.ind and PH-DATA.req start appearing, the jitter is normal and it works nicely.

I have the impression that during that initial face, the FN doesn't advance in the osmo-trx for long periods while it does in osmo-bts-trx (we keep sending stuff). From osmo-bts-trx (the "elapsed\_fn= 0" means the IND CLOCK we received contained same FN than the previous one):

```

20170704122819041 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 185576, elapsed_fn= 0, error_us
=+185576
20170704122819041 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -622844 us (elapsed_fn=-50)
20170704122819041 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819042 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 516, elapsed_fn= 0, error_us
= +516
20170704122819042 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -623360 us (elapsed_fn=-50)

```

```
20170704122819042 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819043 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 1317, elapsed_fn= 0, error_us
=+1317
20170704122819043 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -624678 us (elapsed_fn=-50)
20170704122819043 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819044 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 1329, elapsed_fn= 0, error_us
=+1329
20170704122819045 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -626007 us (elapsed_fn=-50)
20170704122819045 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819045 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 392, elapsed_fn= 0, error_us
= +392
20170704122819045 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -626400 us (elapsed_fn=-50)
20170704122819045 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819045 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 410, elapsed_fn= 0, error_us
= +410
20170704122819045 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -626810 us (elapsed_fn=-50)
20170704122819045 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819046 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 342, elapsed_fn= 0, error_us
= +342
20170704122819046 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -627153 us (elapsed_fn=-50)
20170704122819046 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819046 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 386, elapsed_fn= 0, error_us
= +386
20170704122819046 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -627539 us (elapsed_fn=-50)
20170704122819046 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819047 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 534, elapsed_fn= 0, error_us
= +534
20170704122819047 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -628074 us (elapsed_fn=-50)
20170704122819047 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
20170704122819047 DL1C <0006> scheduler_trx.c:1677 TRX Clock Ind: elapsed_us= 457, elapsed_fn= 0, error_us
= +457
20170704122819047 DL1C <0006> scheduler_trx.c:1695 GSM clock jitter: -628531 us (elapsed_fn=-50)
20170704122819047 DL1C <0006> scheduler_trx.c:1704 We were 50 FN faster than TRX, compensating
```

#21 - 07/04/2017 01:47 PM - laforge

pespin wrote:

The issue seems to be during the initial seconds when both osmo-bts-trx and osmo-trx are powered on and start the configuration phase. Once

the PH-RTS.ind and PH-DATA.req start appearing, the jitter is normal and it works nicely.

Then please establish the details of that early phase. I think

- osmo-trx should not start sending TIME indications via UDP before it is "stable" and able to send those TIME indications in proper intervals
- osmo-bts-trx should not start the internal FN timer until that first TIME message has been received from osmo-trx.

I have the impression that during that initial phase, the FN doesn't advance in the osmo-trx for long periods while it does in osmo-bts-trx (we keep sending stuff). From osmo-bts-trx

why are we at all sending bursts if osmo-trx is not yet ready? I think that's the problem.

#### #22 - 07/04/2017 01:57 PM - pespin

Indeed, I was writing the following in the task but you were faster and wrote before I submitted it:

Enabling info level for DTRX confirms what I was saying: it seems the clock number in IND CLOCK is not updated until we power on the TRX with "CMD POWERON" and "RSP POWERON 0" is received.

When it fails, it does so before receiving the "RSP POWERON 0". Now the question is who's doing it wrong:

- osmo-trx: It should update the clock while powered off (it may make no sense, because it actually it's not transmitting).
- osmo-bts-trx: It should not do clock sanity check during the period in between "CMD POWEROFF" and "RSP POWERON".

I'll do as you suggested. I am actually not sure if we are sending burst or we are only doing the clock sanity checks. I'll look at it.

#### #23 - 07/04/2017 02:05 PM - pespin

I just realized that during the poweroff timeslice (between POWEROFF and RSP POWERON) we basically receive a IND CLOCK just after very "RSP \*\*" message.

**#24 - 07/04/2017 04:35 PM - pespin**

I pushed this osmo-trx commit to Gerrit<sup>1</sup>, it seems to be helping a lot with improving stability of runs in osmo-gsm-tester with osmo-bts-trx.

I think it still doesn't completely fix the issue though, some work needs to be done on the osmo-bts-trx side too. For instance with an earlier version of the osmo-trx patch I once got something like the line below, which is probably saying something's really wrong regarding how we setup the timers:

```
FN timer expire_count=1302: We missed 1301 timers
```

[1] <https://gerrit.osmocom.org/#/c/3120/>

**#25 - 07/05/2017 09:41 AM - pespin**

- Status changed from In Progress to Feedback

Yesterday evening I built osmo-trx used by osmo-gsm-tester with this patch enabled, and since then (built 1146) the osmo-gsm-tester with osmo-bts-trx seems a lot more stable: [http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester\\_run/](http://jenkins.osmocom.org/jenkins/view/osmo-gsm-tester/job/osmo-gsm-tester_run/)

The issues I saw in osmo-bts-trx with the expire\_count may have appeared due to use of previous version of the osmo-trx patch which was sending the first clock indication to early. Let's wait for the osmo-trx patch to be reviewed/merged and see how osmo-gsm-tester evolves. If we see more sporadic failures, then we have some new state and symptoms and I'll spend more time fixing those as a next step.

**#26 - 07/10/2017 06:40 PM - laforge**

- Status changed from Feedback to Stalled

**#27 - 08/25/2017 04:38 PM - pespin**

- Subject changed from sporadic crash of osmo-bts-trx in osmo-gsm-tester runs to sporadic shutdown of osmo-bts-trx in osmo-gsm-tester runs (no clock from trx)

**#28 - 09/14/2017 10:00 AM - pespin**

- File trial-2910-run.tgz added

osmo-bts-trx finally failed with the no-clock issue in production. I attach the run output for that job.

File run.2017-09-14\_01-03-25/sms:trx-b200/mo\_mt\_sms.py/osmo-bts-trx/osmo-bts-trx/stderr contains osmo-bts-trx log with quite a lot of details on timings.

Unfortunately Pcap file in run.2017-09-14\_01-03-25/sms:trx-b200/mo\_mt\_sms.py/osmo-nitb\_10.42.42.2/pcap/ doesn't contain the related GSMTAP due to issues from [#2413](#)

I'll try to have a look at log from osmo-bts-trx soon.



**#29 - 09/26/2017 08:04 AM - pespin**

- File trial-3051-run.tgz added

Another log with debug output giving information on the issue:

run.2017-09-25\_13-05-34/aoip\_sms:trx-b200/mo\_mt\_sms.py/osmo-bts-trx/osmo-bts-trx/stderr

**#30 - 09/26/2017 08:17 AM - pespin**

- File trial-3055-run.tgz added

And another one: run.2017-09-26\_01-05-32/sms:trx-b200/mo\_mt\_sms.py/osmo-bts-trx/osmo-bts-trx/stderr

**#31 - 10/06/2017 10:11 AM - pespin**

- File trial-3184-run.tgz added

One more: aoip\_sms:trx-b200.mo\_mt\_sms.py

**#32 - 10/09/2017 08:53 AM - pespin**

- File trial-3216-run.tgz added

aoip\_sms:trx-b200.mo\_mt\_sms.py

**#33 - 10/09/2017 10:32 AM - pespin**

The remaining no clock issue triggers due to following condition "if (tcs->fn\_without\_clock\_ind++ == TRX\_LOSS\_FRAMES)".

TRX\_LOSS\_FRAMES is 400, and the function `trx_fn_timer_cb` incrementing `fn_without_clock_ind` is called every `FRAME_DURATION_nS=4615384` (4.6ms). Thus, the "no clock" issue is triggered if no CLOCK IND is received after 1.84 seconds ( $400 \cdot 4.6/1000$ ) since previous one.

Looking at different traces from both good and bad runs, it seems generally CLOCK IND are received once per second, even between the 1st and 2nd CLOCK IND. However, in the failing tests, the first CLOCK IND is received and then the condition explained above fires after the 1.84 seconds without receiving the CLOCK IND.

This points so far to an issue in osmo-trx sometimes not sending the 2nd CLOCK IND on time. Investigation is ongoing.

As a rememberer, all this can be calculated from timestamp in messages:  
20171009071158497 DTRX <000b> trx\_if.c:122 Clock indication: fn=89662

Year	Month	Day	Hour	Minute	Second	Milliseconds since start of second
"2017"	"10"	"09"	"07"	"11"	"58"	"497"

I'm getting closer to the issue, which seems to point to osmo-trx / libuhd / B200 HW.

This is the output of osmo-trx during a successful test (CLOCK IND is good):

```
(launched: 2017-10-09_05:26:48.681677)
linux; GNU C++ version 4.9.2; Boost_105500; UHD_003.009.005-0-unknown

opening configuration table from path :memory:
Config Settings
  Log Level..... NOTICE
  Device args.....
  TRX Base Port..... 5700
  TRX Address..... 10.42.42.50
  GSM Core Address.....10.42.42.50
  Channels..... 1
  Tx Samples-per-Symbol... 4
  Rx Samples-per-Symbol... 1
  EDGE support..... Disabled
  Reference..... External
  C0 Filler Table..... Disabled
  Multi-Carrier..... Disabled
  Tuning offset..... 0
  RSSI to dBm offset..... 0
  Swap channels..... 0

-- Detected Device: B200
-- Operating over USB 2.
-- Initialize CODEC control...
-- Initialize Radio control...
-- Performing register loopback test... pass
-- Performing CODEC loopback test... pass
-- Asking for clock rate 16.000000 MHz...
-- Actually got clock rate 16.000000 MHz.
-- Performing timer loopback test... pass
-- Setting master clock rate selection to 'automatic'.
-- Asking for clock rate 26.000000 MHz...
-- Actually got clock rate 26.000000 MHz.
-- Performing timer loopback test... pass
-- Transceiver active with 1 channel(s)
NOTICE 140291600520960 05:26:58.3 Transceiver.cpp:791:driveControl: Changing TSC from 0 to 7
NOTICE 140291600520960 05:26:58.3 Transceiver.cpp:244:start: Starting the transceiver
NOTICE 140291600520960 05:27:39.2 Transceiver.cpp:298:stop: Stopping the transceiver
NOTICE 140291600520960 05:27:39.2 Transceiver.cpp:324:stop: Transceiver stopped
Received shutdown signalShutting down transceiver...
```

And this is the output of a failing test (only 1st CLOCK IND received):

```
(launched: 2017-10-06_07:11:05.739775)
linux; GNU C++ version 4.9.2; Boost_105500; UHD_003.009.005-0-unknown

opening configuration table from path :memory:
Config Settings
  Log Level..... NOTICE
  Device args.....
  TRX Base Port..... 5700
  TRX Address..... 10.42.42.50
  GSM Core Address.....10.42.42.50
  Channels..... 1
  Tx Samples-per-Symbol... 4
  Rx Samples-per-Symbol... 1
  EDGE support..... Disabled
  Reference..... External
  C0 Filler Table..... Disabled
  Multi-Carrier..... Disabled
  Tuning offset..... 0
  RSSI to dBm offset..... 0
  Swap channels..... 0
```

```
-- Detected Device: B200
-- Operating over USB 2.
-- Initialize CODEC control...
-- Initialize Radio control...
-- Performing register loopback test... pass
-- Performing CODEC loopback test... pass
-- Asking for clock rate 16.000000 MHz...
-- Actually got clock rate 16.000000 MHz.
-- Performing timer loopback test... pass
-- Setting master clock rate selection to 'automatic'.
-- Asking for clock rate 26.000000 MHz...
-- Actually got clock rate 26.000000 MHz.
-- Performing timer loopback test... pass
-- Transceiver active with 1 channel(s)
NOTICE 140144875484928 07:11:15.4 Transceiver.cpp:791:driveControl: Changing TSC from 0 to 7
NOTICE 140144875484928 07:11:15.4 Transceiver.cpp:244:start: Starting the transceiver
ERR 140144760469248 07:11:15.6 UHDDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 600509956
ERR 140144760469248 07:11:15.6 UHDDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 374165188
2
ALERT 140144873776896 07:11:15.8 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:15.9 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.0 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.1 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.2 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.3 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.4 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.5 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.6 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.7 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.8 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:16.9 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.0 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.1 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.2 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.3 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.4 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.5 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.6 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.7 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.8 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:17.9 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.0 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.1 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.2 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.3 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.4 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.5 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.6 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.7 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.8 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:18.9 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.0 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.1 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.2 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.3 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.4 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.5 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.6 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.7 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.8 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:19.9 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.0 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.1 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.2 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.3 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.4 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.5 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.6 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.7 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.8 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:20.9 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:21.0 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:21.1 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:21.2 UHDDDevice.cpp:1021:writeSamples: UHD: Device send timed out
```



```
ALERT 140144873776896 07:11:28.9 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:29.0 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:29.1 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
ALERT 140144873776896 07:11:29.2 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
Received shutdown signalShutting down transceiver...
NOTICE 140144875398976 07:11:29.2 Transceiver.cpp:298:stop: Stopping the transceiver
```

This list of errors and ALERTS show up in all failing tests, and doesn't appear in tests running correctly. ALERT seems to be printed every 0.1 seconds, and the time that errors appear in the log matches more a less with the time in which we see issues appearing in osmo-bts-trx log. See the interesting output bits for several recorded wrong tests:

```
NOTICE 140678580717312 13:11:56.2 Transceiver.cpp:791:driveControl: Changing TSC from 0 to 7
NOTICE 140678580717312 13:11:56.2 Transceiver.cpp:244:start: Starting the transceiver
ERR 140678465701632 13:11:56.3 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 235152331
ALERT 140678579009280 13:11:56.5 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
...

NOTICE 139708559849216 01:07:31.3 Transceiver.cpp:244:start: Starting the transceiver
ERR 139708373923584 01:07:31.5 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 369446451
6
ERR 139708373923584 01:07:31.5 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 374165188
2
ALERT 139708558141184 01:07:31.7 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
...

NOTICE 139727734355712 01:11:21.5 Transceiver.cpp:791:driveControl: Changing TSC from 0 to 7
NOTICE 139727734355712 01:11:21.5 Transceiver.cpp:244:start: Starting the transceiver
ERR 139727619340032 01:11:21.7 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 391637379
0
ERR 139727619340032 01:11:21.7 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 374165188
2
ALERT 139727732647680 01:11:21.9 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
...

NOTICE 140144875484928 07:11:15.4 Transceiver.cpp:791:driveControl: Changing TSC from 0 to 7
NOTICE 140144875484928 07:11:15.4 Transceiver.cpp:244:start: Starting the transceiver
ERR 140144760469248 07:11:15.6 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 600509956
ERR 140144760469248 07:11:15.6 UHDDevice.cpp:341:uhd_msg_handler: Got a ctrl packet with unknown SID 374165188
2
ALERT 140144873776896 07:11:15.8 UHDDevice.cpp:1021:writeSamples: UHD: Device send timed out
...
```

The string "Got a ctrl packet with unknown SID 600509956" comes from libuhd, and the numbers for unknown SID point to some kind of memory corruption. On the other hand, SID 3741651882 shows up 3 times, always after the first failed one.

For reference, this is the UHD version showing the failures:

```
# uhd_find_devices type=b200
linux; GNU C++ version 4.9.2; Boost_105500; UHD_003.009.005-0-unknown

-----
-- UHD Device 0
-----
Device Address:
  type: b200
  name:
  serial: 306BD11
  product: B200

# uhd_usrp_probe
linux; GNU C++ version 4.9.2; Boost_105500; UHD_003.009.005-0-unknown

-- Detected Device: B200
-- Operating over USB 2.
```

```
-- Initialize CODEC control...
-- Initialize Radio control...
-- Performing register loopback test... pass
-- Performing CODEC loopback test... pass
-- Asking for clock rate 16.000000 MHz...
-- Actually got clock rate 16.000000 MHz.
-- Performing timer loopback test... pass
-- Setting master clock rate selection to 'automatic'.
```

```
/
| Device: B-Series Device
```

```
| /
| Mboard: B200
```

```
| revision: 4
| product: 1
| serial: 306BD11
| FW Version: 8.0
| FPGA Version: 13.0
```

```
| Time sources: none, internal, external, gpsdo
| Clock sources: internal, external, gpsdo
| Sensors: ref_locked
```

```
| /
| RX DSP: 0
```

```
| Freq range: -8.000 to 8.000 MHz
```

```
| /
| RX Dboard: A
```

```
| /
| RX Frontend: A
```

```
| Name: FE-RX2
| Antennas: TX/RX, RX2
| Sensors: temp, rssi, lo_locked
| Freq range: 50.000 to 6000.000 MHz
| Gain range PGA: 0.0 to 76.0 step 1.0 dB
| Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
| Connection Type: IQ
| Uses LO offset: No
```

```
| /
| RX Codec: A
```

```
| Name: B200 RX dual ADC
| Gain Elements: None
```

```
| /
| TX DSP: 0
```

```
| Freq range: -8.000 to 8.000 MHz
```

```
| /
| TX Dboard: A
```

```
| /
| TX Frontend: A
```

```
| Name: FE-TX2
| Antennas: TX/RX
| Sensors: temp, lo_locked
| Freq range: 50.000 to 6000.000 MHz
| Gain range PGA: 0.0 to 89.8 step 0.2 dB
| Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
| Connection Type: IQ
| Uses LO offset: No
```

```
| /
| TX Codec: A
```

```
| Name: B200 TX dual DAC
| Gain Elements: None
```

**#35 - 10/09/2017 11:24 AM - laforge**

Hi pespin,

On Mon, Oct 09, 2017 at 11:15:10AM +0000, pespin [REDMINE] wrote:

I'm getting closer to the issue, which seems to point to osmo-trx / B200 HW.

In fact, it points at some osmo-trx/UHD interaction.

Please raise this as a separate bug in the osmo-trx project and add ttsou as an explicit watcher and/or ping him by e-mail.

**#36 - 10/29/2017 06:51 PM - laforge**

- Priority changed from Normal to Low

**#37 - 10/29/2017 06:55 PM - laforge**

- Target version set to osmo-bts-trx refresh

**#38 - 11/09/2017 11:37 AM - pespin**

It seems sporadic failures also happen in osmo-bts-trx in RnD setup running debian 9.

On a normal run, stderr output in osmo-trx is empty. However, I see the following output in osmo-trx when the issue happens:

```
(launched: 2017-11-09_12:30:29.792747)
ERR 139864736507648 12:30:51.2 UHDDevice.cpp:1382:write: Skipping buffer data: timestamp=3879316 time_end=3856
832
ERR 139864736507648 12:30:51.3 UHDDevice.cpp:843:check_rx_md_err: No packet received, implementation timed-out
ALERT 139864736507648 12:30:51.3 UHDDevice.cpp:847:check_rx_md_err: UHD: Receive timed out
ALERT 139864736507648 12:30:51.3 radioInterface.cpp:320:pullBuffer: Receive error 0
ERR 139864736507648 12:30:51.3 UHDDevice.cpp:1382:write: Skipping buffer data: timestamp=3938263 time_end=3882
628
```

```
# aptitude show libuhd003
Package: libuhd003
Version: 3.9.5-2+b3
State: installed
Automatically installed: yes
Multi-Arch: same
Priority: optional
Section: libs
Maintainer: A. Maitland Bottoms <bottoms@debian.org>
Architecture: amd64
Uncompressed Size: 7,553 k
Depends: libboost-atomic1.62.0, libboost-chrono1.62.0, libboost-date-time1.62.0,
libboost-filesystem1.62.0, libboost-program-options1.62.0, libboost-regex1.62.0,
libboost-serialization1.62.0, libboost-system1.62.0, libboost-test1.62.0,
libboost-thread1.62.0, libc6 (>= 2.15), libgcc1 (>= 1:3.0), libstdc++6 (>= 5.2),
libusb-1.0-0 (>= 2:1.0.9)
Suggests: gnuradio
```

```
Breaks: gnuradio (< 3.7.8-3), libgnuradio-osmosdr0.1.3 (< 0.1.3-3), libgnuradio-uhd3.5.3.2 (< 3.6), libgnuradio-uhd3.7.5 (< 3.7.8), gnuradio:i386 (< 3.7.8-3), libuhd003:i386 (!= 3.9.5-2+b3)
```

```
Replaces: libuhd003:i386 (< 3.9.5-2+b3)
```

```
Description: universal hardware driver for Ettus Research products - library  
Host library for the Universal Hardware Driver for Ettus Research products.
```

The supported devices provide analog radio receiver and transmitter hardware along with digital interfaces for getting signals to and from a software defined radio running on the host computer.

```
Homepage: http://www.ettus.com/sdr-software/detail/usrp-hardware-driver
```

```
Tags: role::shared-lib
```

```
root@osmo-gsm-tester-rnd:/var/tmp/osmo-gsm-tester/trials/trial# uhd_find_devices type=b200  
linux; GNU C++ version 6.3.0 20170221; Boost_106200; UHD_003.009.005-0-unknown
```

```
-----  
-- UHD Device 0  
-----
```

```
Device Address:
```

```
type: b200  
name: MyB200  
serial: 30A9FFB  
product: B200
```

```
root@osmo-gsm-tester-rnd:/var/tmp/osmo-gsm-tester/trials/trial# uhd_usrp_probe  
linux; GNU C++ version 6.3.0 20170221; Boost_106200; UHD_003.009.005-0-unknown
```

```
-- Detected Device: B200  
-- Operating over USB 2.  
-- Initialize CODEC control...  
-- Initialize Radio control...  
-- Performing register loopback test... pass  
-- Performing CODEC loopback test... pass  
-- Asking for clock rate 16.000000 MHz...  
-- Actually got clock rate 16.000000 MHz.  
-- Performing timer loopback test... pass  
-- Setting master clock rate selection to 'automatic'.
```

```
/-----  
| Device: B-Series Device  
|-----  
| /  
| | Mboard: B200  
| | revision: 5  
| | product: 1  
| | serial: 30A9FFB  
| | name: MyB200  
| | FW Version: 8.0  
| | FPGA Version: 13.0  
| |  
| | Time sources: none, internal, external, gpsdo  
| | Clock sources: internal, external, gpsdo  
| | Sensors: ref_locked  
| |-----  
| | /  
| | | RX DSP: 0  
| | | Freq range: -8.000 to 8.000 MHz  
| | |-----  
| | | /  
| | | RX Dboard: A  
| | |-----  
| | | /  
| | | | RX Frontend: A  
| | | | Name: FE-RX1  
| | | | Antennas: TX/RX, RX2  
| | | | Sensors: temp, rssi, lo_locked  
| | | | Freq range: 50.000 to 6000.000 MHz  
| | | | Gain range PGA: 0.0 to 76.0 step 1.0 dB  
| | | | Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz  
| | | | Connection Type: IQ  
| | | | Uses LO offset: No  
| | | |-----  
| | | /  
| | | RX Codec: A
```



```

| | | | Name: B200 RX dual ADC
| | | | Gain Elements: None
| | | |
| | | | /
| | | | TX DSP: 0
| | | | Freq range: -8.000 to 8.000 MHz
| | | |
| | | | /
| | | | TX Dboard: A
| | | |
| | | | /
| | | | TX Frontend: A
| | | | Name: FE-TX1
| | | | Antennas: TX/RX
| | | | Sensors: temp, lo_locked
| | | | Freq range: 50.000 to 6000.000 MHz
| | | | Gain range PGA: 0.0 to 89.8 step 0.2 dB
| | | | Bandwidth range: 200000.0 to 56000000.0 step 0.0 Hz
| | | | Connection Type: IQ
| | | | Uses LO offset: No
| | | |
| | | | /
| | | | TX Codec: A
| | | | Name: B200 TX dual DAC
| | | | Gain Elements: None

```

**#39 - 09/30/2018 11:12 AM - laforge**

is this still happening?

**#40 - 09/30/2018 10:07 PM - pespín**

I think so. Let's see if it persists once we move to using the i5 machine to run osmo-trx.

## Files

File Name	Size	Date	Owner
trial-2910-run.tgz	1.63 MB	09/14/2017	pespin
trial-3051-run.tgz	1.78 MB	09/26/2017	pespin
trial-3055-run.tgz	1.81 MB	09/26/2017	pespin
trial-3184-run.tgz	1.77 MB	10/06/2017	pespin
trial-3216-run.tgz	1.79 MB	10/09/2017	pespin