OsmoGSMTester - Feature #2760

osmo-gsm-tester: Add support for several (osmo-)trx to OsmoBtsTrx

12/15/2017 12:30 PM - pespin

Status:	Closed	**Start date:**	12/15/2017
Priority:	Normal	**Due date:**	
Assignee:	pespin	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

Right now we only have 2 TRX support for OctoBTS, and we are actually not using it yet

We need to find out which is the best way to implement the API to select and use several TRX depending on the test.
See https://gerrit.osmocom.org/#/c/4674/ for related discussion. Summary: Have a generic BTS API with something like set_enabled_trx(NUM), get_enabled_trx(), which sets a enabled_trx variable which is used at conf_for_bsc() time to only pass "enabled_trx" amount of trx in trx_list. This can be configured by default too using a "enabled_trx" config option for each bts in resources.conf.

We also lack multiple TRX support in osmo-bts-trx. It should not be difficult to add support for it. We basically require to launch a new osmo-trx process (if launch_trx option is enabled) for each TRX configured.

We can then look at bts_octphy.py to see which code we can share/merge (move into OsmoBtsMainUnit) that handles several TRX. Octphy has some specific constrains so we may not be able to share everything, as it requires to set up the phys and instances in a given way depending on hw_addr.

Related issues:

Related to OsmoGSMTester - Bug #2761: osmo-gsm-tester: add test case: Test 2n...	**Resolved**	**12/15/2017**
Related to OsmoGSMTester - Bug #3560: nanoBTS multiTRX tests in osmo-gsm-test...	**Stalled**	**09/17/2018**

History

#1 - 12/15/2017 12:37 PM - pespin

- Related to Bug #2761: osmo-gsm-tester: add test case: Test 2nd trx is correctly used added

#2 - 12/15/2017 02:06 PM - laforge

On Fri, Dec 15, 2017 at 12:30:47PM +0000, pespin [REDMINE] wrote:

> We also lack multiple TRX support in osmo-bts-trx.

I presume you want to say "in the osmo-gsm-tester support for osmo-bts-trx" ?

> It should not be difficult to add support for it. We basically require to launch a new osmo-trx process (if launch_trx option is enabled) for each TRX configured.

The normal method used so far AFAIK is to run two soft-TRXs within one osmo-trx instance. This will create two ARFCN carriers inside the spectrum of the USRP. That's what the "-c 2" option of osmo-trx would do. This is similar to osmo-bts-lc15 or osmo-bts-octphy where we run two phy_instances in one phy_link.

I'm not sure if the "run one phy_instance in each phy_link of two separate osmo-trx instances" would work without significant changes to osmo-trx, whcih I believe is out of scope here.

#3 - 12/15/2017 02:46 PM - pespin

laforge wrote:

> On Fri, Dec 15, 2017 at 12:30:47PM +0000, pespin [REDMINE] wrote:

> > We also lack multiple TRX support in osmo-bts-trx.
>
> I presume you want to say "in the osmo-gsm-tester support for osmo-bts-trx" ?

yes of course :-) I was writing that having osmo-gsm-tester context in mind.

> The normal method used so far AFAIK is to run two soft-TRXs within one
> osmo-trx instance. This will create two ARFCN carriers inside the
> spectrum of the USRP. That's what the "-c 2" option of osmo-trx would
> do. This is similar to osmo-bts-lc15 or osmo-bts-octphy where we run
> two phy_instances in one phy_link.
>
> I'm not sure if the "run one phy_instance in each phy_link of two
> separate osmo-trx instances" would work without significant changes to
> osmo-trx, whcih I believe is out of scope here.

Ah thanks for the clarification, I wrote everything from what I remembered at the time off the top of my head. Then supporting it (several phy instances per phy device) should be easier than expected then. Let's target for that one then. However, I'm not sure if we actually have any osmo-trx devices that works with more than 1 trx. AFAIK sysmocell5000 doesn't support ore than 1, and same goes for B200. The LimeSDR (not the mini one) theoretically could support 2 TRX.

#4 - 12/15/2017 06:29 PM - laforge

On Fri, Dec 15, 2017 at 02:46:01PM +0000, pespin [REDMINE] wrote:

> However, I'm not sure if we actually have any osmo-trx devices that
> works with more than 1 trx. AFAIK sysmocell5000 doesn't support ore
> than 1, and same goes for B200. The LimeSDR (not the mini one)
> theoretically could support 2 TRX.

Correct for sysmoCell 5000. But for B200 and Lime:

I'm sometimes surprised where you get come up with those
assumptions/presumptions ;) This is not a rhethoric question! I'm
really curious, where you found that information, because it's wrong.
So let's try to find the source of this and fix it to avoid other people
drawing the same conclusions.

osmo-trx has had two-TRX suport on USRPs since.. I think forever. Even
OpenBTS Transceiver from which it is inherited had the feature, AFAIR. At
least the "-c" option was already present in the first commit in 2011,
so I assume it was already working back then.

Also, why would LimeSDR and LimeSDR mini be different? It's the same
transceiver chip, isn't it?

#5 - 12/15/2017 10:01 PM - pespin

laforge wrote:

> Correct for sysmoCell 5000. But for B200 and Lime:
>
> I'm sometimes surprised where you get come up with those
> assumptions/presumptions ;) This is not a rhethoric question! I'm
> really curious, where you found that information, because it's wrong.
> So let's try to find the source of this and fix it to avoid other people
> drawing the same conclusions.

It's more like the opposite: I don't remember reading any information in any place stating that 2 TRX are supported, and I don't remember neither hearing about somebody operating it with more than 1 TRX, so my assumption is that it's most probably not supported, but I see I assumed wrong in this case :-)

> Also, why would LimeSDR and LimeSDR mini be different? It's the same
> transceiver chip, isn't it?

I recall seeing it when doing a quick look at https://www.crowdsupply.com/lime-micro/limesdr-mini (search for "Comparison Table"). It states LimeSDR has 2 TX and 2 RX channels while for LimeSDR Mini it states 1+1. Did I understand that information incorrectly?
Actually, now that I look closer at it, it seems to be that Ettus B200 supports only 1 while Ettus B210 supports 2.

#6 - 12/16/2017 11:10 AM - laforge

On Fri, Dec 15, 2017 at 10:01:05PM +0000, pespin [REDMINE] wrote:

> > Also, why would LimeSDR and LimeSDR mini be different? It's the same
> > transceiver chip, isn't it?
>
> I recall seeing it when doing a quick look at https://www.crowdsupply.com/lime-micro/limesdr-mini (search for "Comparison Table"). It states LimeSDR has 2 TX and 2 RX channels while for LimeSDR Mini it states 1+1. Did I understand that information incorrectly?
> Actually, now that I look closer at it, it seems to be that Ettus B200 supports only 1 while Ettus B210 supports 2.

The count of physical connectors on a given board has absolutely no relationship to the question of how many individual GSM RF carriers you are putting on it.

Let's say you have a 5 MHz wide transmit/receive band at the ADC/DAC of your SDR. Within those 5 MHz you can put a number of the only 270kHz wide GSM carriers. In theory up to 9 non-overlapping ARFCNs (i.e. with one ARFCN space in between), but that's unrealistic/impractical for a large number of other factors, last but not least computational complexity.

#7 - 08/23/2018 04:13 PM - pespin

Side note: See https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/10583/ as an example on how to set up osmo-gsm-tester config to use more than 1 TRX.

#8 - 09/17/2018 01:55 PM - laforge

what's that status here? aside from the "side note" 25 days ago, there hasn't been any update for 9 months?

#9 - 09/17/2018 02:10 PM - pespin

Status is: We have a reasonably good way to define and configure several TRXs now in osmo-gsm-tester, and it has been tested with nanobts. What is lacking is implementing and testing the related bits in ./src/osmo_gsm_tester/bts_osmo.py to generate the osmo-bts-trx .cfg file with those TRX, and pass the related config to osmo-bsc. I guess we also need to configure osmo-trx launched accordingly (1 channel per TRX?).

I can work on that soon, shall I increase priority?

#10 - 09/17/2018 02:30 PM - laforge

- Related to Bug #3560: nanoBTS multiTRX tests in osmo-gsm-tester Prod setup failing added

#11 - 09/17/2018 06:34 PM - pespin

- Status changed from New to In Progress

- % Done changed from 0 to 60

Support added in:
https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/11000 osmo-bts-trx: Add multiTRX support
https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/11001 osmo-trx: Add multi_arfcn support

However, we cannot enable it yet for B200 because that on can only do multiTRX through multi_arfcn and the APU machine we use to run osmo-gsm-tester cannot cope with CPU requirements added by multi_arfcn overhead. osmo-bts-trx and osmo-trx will fail after a few seconds due to timers failing to do stuff on time.

roh , most probably we'll need to upgrade it to an APU2. Maybe we should do some manual tests with one to make sure an APU2 is enough beforehand too.

Tomorrow I'll give a try to sysmocell5000 to see if it supports configuration of 2 TRX.

#12 - 09/18/2018 03:39 PM - pespin

- Assignee changed from pespin to roh

Assigning to roh for him to set up a core network with an ettus B200 in an APU2 with osmo-trx running with multi-arfcn enabled to see if the APU2 is able to keep up with the CPU load.

Sample osmo-trx.cfg (as used by osmo-gsm-tester [1]) with multi-arfcn enabled: (you probably need to run as root due to rt-prio 18)

```
!
! OsmoTRX example configuration
!
log stderr
  logging filter all 1
  logging color 1
  logging print category 1
  logging timestamp 1
  logging print file basename
  logging print extended-timestamp 1
  logging level all info
!
line vty
 bind 127.0.0.1
ctrl
 bind 127.0.0.1
trx
 bind-ip 127.0.0.1
 remote-ip 127.0.0.1
 base-port 5700
 egprs disable
 multi-arfcn enable
 tx-sps 4
 rx-sps 4
 clock-ref internal
 rt-prio 18
 chan 0
 chan 1
```

And osmo-bts-trx [2]:

```
! Configuration rendered by osmo-gsm-tester
log stderr
  logging color 1
  logging print extended-timestamp 1
  logging print category 1
  logging level abis debug
  logging level oml debug
  logging level pag debug
  logging level rll debug
  logging level rr debug
  logging level rsl debug
  logging level l1c info
  logging level l1p error
  logging level trx info
  ! Level required by ready_for_pcu(): pcu info
  logging level pcu info
!
line vty
 bind 127.0.0.1
ctrl
 bind 127.0.0.1
!
phy 0
 osmotrx ip local 127.0.0.1
 osmotrx ip remote 127.0.0.1
 instance 0
  osmotrx rx-gain 25
  osmotrx tx-attenuation oml
 instance 1
  osmotrx rx-gain 25
  osmotrx tx-attenuation oml
bts 0
 band GSM-1800
 ipa unit-id 6 0
 oml remote-ip 127.0.0.1
 pcu-socket /tmp/foo
 gsmtap-sapi bcch
 gsmtap-sapi ccch
 gsmtap-sapi rach
 gsmtap-sapi agch
 gsmtap-sapi pch
```

```
 gsmtap-sapi sdcch
 gsmtap-sapi tch/f
 gsmtap-sapi tch/h
 gsmtap-sapi pacch
 gsmtap-sapi pdtch
 gsmtap-sapi ptcch
 gsmtap-sapi cbch
 gsmtap-sapi sacch
 trx 0
  phy 0 instance 0
 trx 1
  phy 0 instance 1
```

In osmo-bsc, you need to set the arfcn for each TRX with a difference of 4. In osmo-gsm-tester case we use 868 and 872.

[1] https://git.osmocom.org/osmo-gsm-tester/tree/src/osmo_gsm_tester/templates/osmo-trx.cfg.tmpl
[2] https://git.osmocom.org/osmo-gsm-tester/tree/src/osmo_gsm_tester/templates/osmo-bts-trx.cfg.tmpl

#13 - 09/19/2018 11:54 AM - roh

- File usrp_multi_trx_2.gif added

- File usrp_multi_trx_1.gif added

- Assignee changed from roh to pespin

i did some tests now and compared multi-arfcn on apu2 to only one arfcn.

one arfcn: one core at ~70% load. system working nicely.

multi-arfcn: one core at 100%, one at 70%, one at 30 and one at 25. the osmo-trx-uhd process shows 250-260% - lots of skipping buffers, system overloaded.

Wed Sep 19 13:31:11 2018 DDEV <0001> UHDDevice.cpp:857 [tid=140404717168384] An internal receive buffer has filled at 307.432 sec.
Wed Sep 19 13:31:11 2018 DDEV <0001> UHDDevice.cpp:1481 [tid=140404717168384] Skipping buffer data: timestamp=983858669 time_end=983783547
Wed Sep 19 13:31:11 2018 DMAIN <0000> Transceiver.cpp:1037 [tid=140404717168384] ClockInterface: sending IND CLOCK 1253646
---

also the osmo-bts-trx process keeps dying:
<0007> l1sap.c:463 1285625/969/03/17/41 Invalid condition detected: Frame difference is 1285625-1285559=66 > 1!

---

i also checked the load per thread:
single arfcn:
4 threads, all below 25% cpu

multi arfcn:
6? threads, one 100%, one 80% 2x 25% and 2x 15%

maybe the load could be even more threaded, but i doubt that this will be enough to get this working with an apu2.

i guess we'll need some i3 or i5 to do this kind of math - if it cannot be optimized a bunch more.

also the signal slope of the 2nd trx looks quite bad, but i am not sure if that changes with enough cpu. also the bts was crashing so fast it was difficult to capture.

note: osmo-bts-trx needs the parameter -t 2 to start

#14 - 09/19/2018 02:10 PM - laforge

Ok, I've ordered a supermicro 1U server with front-io (connectors all to front)
in the following configuration:

- CPU: Intel XEON D-2123IT / 4C/8T - 2,20GHz / 8MB / 60W
- Motherboard: Supermicro X11SDV-4C-TLN2F / 2x10Gbit RJ-45 / 1x NVMe / 4 DIMM / IPMI
- Arbeitsspeicher: 8GB / 1x8GB / ECC registered DDR4-2400/2666 / RDIMM
- 1. Festplatte: SSD SATA 6Gb/s - 250 GB - 150 TBW - Samsung Consumer 860 Evo MZ-76E250B

The CPU should be slightly faster (for single and multi thread case) than the
i7-6500U that I have in my rather high-end Thinkpad x260, so I'm confident we
should be safe there. Adding more cores didn't really seem to make sense for
such a special-purpose system.

The enclosure is only 25cm deep, so it fits nicely into our LTHW rack.

#15 - 10/02/2018 02:28 PM - pespin

- File trial-155-run.tgz added

- % Done changed from 60 to 70

I added a few commits to osmo-gsm-tester to be able to run osmo-trx in a remote machine through ssh (took more than expected due to issues with killing process after ssh session is stopped).

https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/11191/ osmotrx: Allow running osmo-trx from remote host
https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/11192/ osmotrx: Make sure remote process stops after ssh session is closed
https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/11196 resources.conf.prod: Use specific remote machine to run osmo-trx
https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+/11197 osmo-trx: Enable multi_arfcn for B200 and only in multiTRX setup

I then configured the new machine and run some tests to check how it behaves. With 1 TRX is runs nicely and tests pass. However, when running with multi-arfcn enabled and 2 TRX, the MS is able to see the network but registering fails with status=denied.
This is how I'm running it (with osmo-gsm-tester branch pespin/remote-trx): "-s ussd:trx-b200+mod-bts0-numtrx2+mod-bts0-chanallocdescend"
Looking more in detail at traces (see attached run.tgz), GSMTAP shows all the usual System Information downlink messages, and I can even see an uplink RACH request (gsmtap.uplink == 1), but nothing else.

In logs I see lots of messages with an always increasing by 1 value (starting from 0:4):

```
...
[0;m20181002160541813 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4958
[0;m20181002160541866 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4959
[0;m20181002160541976 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4960
[0;m20181002160542026 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4961
[0;m20181002160542066 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1037 [tid=140021839468288] ClockInterface: sendin
g IND CLOCK 386945
[0;m20181002160542075 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4962
[0;m20181002160542126 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4963
[0;m20181002160542180 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4964
[0;m20181002160542226 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4965
[0;m20181002160542279 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4966
[0;m20181002160542334 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4967
[0;m20181002160542382 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4968
[0;m20181002160542436 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4969
[0;m20181002160542484 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4970
[0;m20181002160542539 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4971
[0;m20181002160542591 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4972
[0;m20181002160542655 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4973
[0;m20181002160542707 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4974
[0;m20181002160542822 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4975
[0;m20181002160542877 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4976
[0;m20181002160542935 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4977
[0;m20181002160542986 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4978
[0;m20181002160543053 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1037 [tid=140021839468288] ClockInterface: sendin
g IND CLOCK 387162
[0;m20181002160543083 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4979
[0;m20181002160543140 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4980
[0;m20181002160543187 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4981
[0;m20181002160543239 [1;32mDMAIN[0;m <0000> Transceiver.cpp:1004 [tid=140021839501056] new latency: 0:4982
```

#16 - 10/02/2018 02:33 PM - pespin

We are entering this condition every time (Transceiver.cpp:1004):

```
      // if underrun, then we're not providing bursts to radio/USRP fast
      //   enough.  Need to increase latency by one GSM frame.
      if (mRadioInterface->getWindowType() == RadioDevice::TX_WINDOW_USRP1) {
        if (mRadioInterface->isUnderrun()) {
          // only update latency at the defined frame interval
          if (radioClock->get() > mLatencyUpdateTime + GSM::Time(USB_LATENCY_INTRVL)) {
            mTransmitLatency = mTransmitLatency + GSM::Time(1,0);
            LOG(INFO) << "new latency: " << mTransmitLatency;
            mLatencyUpdateTime = radioClock->get();
          }
        }
...
```

#17 - 10/03/2018 02:01 PM - pespin

From BTS point of view, one RACH request is received but it's discarded due to bad quality:

```
20181002160512715 DL1C <0006> l1sap.c:1234 380582/287/20/20/10 Ignoring RACH request: BER10k(1944) > BER10k_MA
X(1707)
```

#18 - 11/01/2018 04:57 PM - pespin

I tested B200 with 2 chan and multi-arfcn on my local setup, and it's working fine there (see #3475).

#19 - 09/20/2019 10:29 AM - pespin

- Status changed from In Progress to Closed

- % Done changed from 70 to 100

osmo-gsm-tester already supports running multi-trx and multi-arfcn. b200 multi-arfcn osmo-trx issues are being tracked currently in #4207, so let's close this ticket.

Files

usrp_multi_trx_2.gif	18.7 KB	09/19/2018	roh
usrp_multi_trx_1.gif	17.3 KB	09/19/2018	roh
trial-155-run.tgz	207 KB	10/02/2018	pespin