On a setup with current nightly osmo-trx-uhd + osmo-bts-trx, dexter and I am observing:

```<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 1440 vs 160 (934964->935003)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 82594400 vs 160 (934999->935038)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 1440 vs 160 (935007->935045)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 82594400 vs 160 (935045->935082)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 1440 vs 160 (935082->935119)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 82594400 vs 160 (935119->935156)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 1440 vs 160 (935156->935193)
<000e> l1sap.c:143 (bts=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 82594400 vs 160 (935193->935230)
<000e> scheduler_trx.c:1780 TRX Clock Ind: elapsed_us=1000033, elapsed_fn=217, error_us=-1422
<000e> scheduler_trx.c:1798 GSM clock jitter: -1944us (elapsed_fn=0)
<000e> l1sap.c:143 (bts=0,trx=0,ts=2,ss=0) RTP clock out of sync with lower layer: 1440 vs 160 (935189->935228)
```

Description

OsmoBTS - Bug #4461

RTP clock out of sync with lower layer

03/19/2020 08:41 PM - laforge

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
</tr>
<tr>
<td>Assignee:</td>
<td>dexter</td>
</tr>
<tr>
<td>Category:</td>
<td>osmo-bts-trx</td>
</tr>
</tbody>
</table>

Spec Reference: 

On a setup with current nightly osmo-trx-uhd + osmo-bts-trx, dexter and I am observing:
What is obvious is that

- it's always the magic values 1440 / 82594400
  - 1440 corresponds to 9 blocks of 160 clocks of our 8kHz sample clock
- frame numbers are jumping back and forth
this means we actually receive TCH data first for FN 935003, then advance to 935029 after which we receive TCH data for 934999. This is impossible and clearly indicates some problem in our scheduler.

Related issues:
Related to OsmoBTS - Feature #3428: Implement handling of NOPE / IDLE indicat... Resolved 07/28/2018

History
#1 - 03/19/2020 08:42 PM - laforge
- Category set to osmo-bts-trx

#2 - 03/19/2020 09:31 PM - laforge
- % Done changed from 0 to 10

If I deactivate the 'handle loss detection of received TCH frame' sectio n from trx_tch_common(), The frame number jumps "into the future" are gone. It seems like that code is injecting uplink TCH.ind with frame numbers of the downlink. And downlink is of course advanced by "fn-advance" and hence wrong.

I still see plenty of those on a bad radio link:

If I deactivate the 'handle loss detection of received TCH frame' sectio n from trx_tch_common(), The frame number jumps "into the future" are gone. It seems like that code is injecting uplink TCH.ind with frame numbers of the downlink. And downlink is of course advanced by "fn-advance" and hence wrong.

I still see plenty of those on a bad radio link:
l1sap.c:143 (bps=0,trx=0,ts=1,ss=0) RTP clock out of sync with lower layer: 320 vs 160 (2460297->2460300)
05/16/2020 4/6
which means that somehow we are not sending a "bad frame" TCH.ind up the L1SAP?

#3 - 03/19/2020 09:42 PM - laforge

- Related to Feature #3428: Implement handling of NOPE / IDLE indications from Transceiver added

#5 - 03/20/2020 12:21 PM - dexter

As discussed earlier this day I have now removed the nope_fn handler. The ul_fn handler is now called each time a nope indications arrives. This ensures proper clocking in osmo-bts-trx and the dropout handler in the scheduler should become active anymore.

https://gerrit.osmocom.org/c/osmo-bts/+/17539 scheduler: always call ul handler on nope ind

#6 - 03/22/2020 01:09 PM - laforge

See https://gerrit.osmocom.org/c/osmo-bts/+/17566 and https://gerrit.osmocom.org/c/osmo-bts/+/17567

#7 - 03/25/2020 08:41 PM - laforge

- Assignee changed from sysmocom to d8g11qrrlqin
- Priority changed from Urgent to High
- % Done changed from 10 to 90

setting this to 90% and assinging to dexter. The only missing part is the follow-up patch to remote the nope_fn function pointer (And make sure all functions like rx_rach_fn can deal with burst_len==0).

#8 - 03/25/2020 08:41 PM - laforge
- Assignee changed from d8g11qnlqin to dexter

#9 - 04/20/2020 09:28 AM - dexter
- Status changed from New to Resolved
- % Done changed from 90 to 100

Change lce45d5986610d9bcefl2a7e41f0a395ec779e3928 is merged, so this can be closed now