

OsmoBTS - Feature #1776

Implement fully dynamic TCH/F + TCH/H + PDCH switching

07/12/2016 05:13 PM - laforge

Status: Closed	Start date: 07/12/2016
Priority: Urgent	Due date:
Assignee: neels	% Done: 100%
Category:	
Target version: Dynamic TCH/H TCH/F PDCH	
Spec Reference:	
Description	
Following the plan of #1757 , please implement fully dynamic switching. Initial target is osmo-bts-sysmo and omso-bts-lc15.	
Related issues:	
Related to OsmoBTS - Bug #1564: dynamic TCH/F + TCH/H allocation	Closed 02/23/2016
Related to OsmoMSC - Feature #1778: avoid mismatching TCH/F vs TCH/H pchan types	Resolved 07/21/2016
Related to OsmoBTS - Bug #1796: PTCH activation breaks dyn TS	Closed 08/10/2016

History

#1 - 07/13/2016 01:29 PM - neels

I thought [#1564](#) was about this?

#2 - 07/13/2016 05:27 PM - neels

(copying from [#1757](#))

Things to be done and working hours needed as estimated:

1. Adjust general structures (1h)

1.1 enum gsm_phys_chan_config: GSM_PCHAN_DYN

1.2 struct gsm_lchan: state machine for chan type transitions

2. openbsc (12h total, optional +16h)

2.1 initialization as PDCH: question: implicit or explicit?

- For reference: in ip.access TCH/F_PDCH, default type is TCH/F, and nm_statechg_event() sends PDCH ACT immediately after init. We could do the same: in the BTS code, initialize as TCH/F, and in nm_statechg_event() trigger a channel state transition to PDCH.
- This feels very heavy on messages though, we could just imply on the BTS side that a DYN channel always starts off as PDCH --> no further initialization messages needed from the BSC side.
- How does the ericsson BTS expect this?

2.2 lchan_alloc(): pick and switch dyn channels (4h)

The initial strategy for picking channels will be fairly straightforward in that a TS will be picked iff it is fully available. More complex scenarios may be solved by implementing the TS defragmentation as in 2.4 below.

- pick matching dedicated pchan first if available
- TCH/H: pick second TCH/H if such TS is used only by half
- otherwise pick DYN TS that is in PDCH mode and switch to TCH/*

2.3 channel type transition steps from BSC perspective (8h)

- BTS indicates the need for a TCH/* channel
- rsl_rx_chan_rq() calls lchan_alloc() and gets a DYN channel in PDCH mode
- rsl_rx_chan_rq() then invokes rsl_chan_activate_lchan()

- rsl_chan_activate_lchan(): detect need to change type, send chan rel
- BTS releases channel
- rsl_rx_rf_chan_rel_ack(): register release, send chan act in TCH/* type
- BTS activates channel
- rsl_rx_chan_act_ack(): register act, continue in rsl_chan_activate_lchan()
- TCH/* is used...
- BTS sends rel ind as usual
- BSC sends chan rel as usual
- rsl_rx_rf_chan_rel_ack(): notices a DYN channel is released in TCH/* mode
 - If half a TCH/H TS is still used, do not continue with PDCH act
- rsl_rx_rf_chan_rel_ack(): sends chan act in PDCH type
- rsl_rx_chan_act_ack(): register act as PDCH, DYN available for lchan_alloc()

2.4 Optional: TS defragmentation (16h)

- Upon each channel release, do handover within TRX to eliminate gaps
- There is some untested defrag code adjusted from jolly in branch neels/dyn_pdch_extra
- Possibly combine two half-used TCH/H TS

3. osmo-bts common implementation (8h total)

3.2 channel type switching: implicit TS disconnect (4h)

- rsl_rx_chan_rel(): usual release will lead to l1sap_info_rel_cnf()
- l1sap_info_rel_cnf(): disconnect DYN TS in L1, to be able to switch type
 - use new bts_model_ts_disconnect(), expecting disconnect callback (dyn_pdch_ts_disconnected(), probably rename to dyn_ts_disconnected)
 - in callback, continue l1sap_info_rel_cnf() to send RSL_RF_CHAN_REL ACK

3.3 channel type switching: implicit TS connect (4h)

- rsl_rx_chan_activ(): first connect DYN TS in different type
 - use new bts_model_ts_connect() with callback (dyn_pdch_ts_connected(), probably rename to dyn_ts_connected)
 - in callback, continue with rsl_rx_chan_activ() procedure leading to RSL_CHAN_ACTIV ACK

4. BTS model specific implementations (6h total + optional 4h)

- Implement:
 - bts_model_opstart(): start DYN channels in default mode (PDCH? see above)
 - ensure bts_model_ts_connect() and bts_model_ts_disconnect() are implemented
- on these models:
 - sysmoBTS (1h, because bts_model_ts_* already implemented)
 - litecell15 (1h, because bts_model_ts_* already implemented)
 - osmo-trx (4h)
 - octphy? (4h)

5. Testing (19.5h + optional 6.5h)

- Test these things (1.5h / 2.5h)
 - a) GPRS initially working? (.5h)
 - b) TCH/F voice working? (.5h)
 - c) TCH/H voice working? (.5h)
 - d) PDCH TS still working after it switched back from TCH/*? (.5h)
 - e) dedicated pchan preferred over DYN TS? (.5h)
- In these scenarios (6.5h)
 - 1 No DYN channels (test against regressions) (a-c: 1.5h)
 - 2 All channels as DYN (a-e: 2.5h)
 - 3 realistic: few dedicated PDCH, all others DYN (a-e: 2.5h)
- On these models
 - sysmoBTS
 - litecell15
 - osmo-trx
 - octphy?

6. Effort estimation

Summing up, I have

1. 1h
2. 12h + 16h TS defrag

3. 8h
4. 6h + 4h octphy
5. 19.5h + 6.5h octphy

Some things may not work as expected or take longer to implement, thus I would like to add a grace of three days (+24h). Highest uncertainty in decreasing order: octphy, osmo-trx, TS defrag, general testing, channel allocator.

Total: 46.5h + 24h grace + 16h TS defrag + 10.5h octphy

My final estimation is thus two of my working weeks (70.5h) plus two days for TS defrag and another generous day for octphy, say three weeks for the whole package.

I hope this estimate makes sense to you. It is hard to estimate software engineering progress in advance; while I don't want to stop the show with huge numbers, it usually turns out realistic to estimate generously.

I am mildly optimistic in this case since I'm familiar with most of the implementations from the recent addition of IPAC_PDCH_[DE]ACT on ip.access nanobts, SysmoBTS and litecell15.

OTOH I am not yet familiar with osmo-trx at all and have minimal experience with octphy. I would still need to set up a tool chain and test environment for both.

#3 - 07/13/2016 05:33 PM - neels

neels wrote:

2.1 initialization as PDCH: question: implicit or explicit?

- For reference: in ip.access TCH/F_PDCH, default type is TCH/F, and nm_statechg_event() sends PDCH ACT immediately after init. We could do the same: in the BTS code, initialize as TCH/F, and in nm_statechg_event() trigger a channel state transition to PDCH.
- This feels very heavy on messages though, we could just imply on the BTS side that a DYN channel always starts off as PDCH --> no further initialization messages needed from the BSC side.
- How does the ericsson BTS expect this?

The channel-activation message has been expanded to include a PDCH type. So I expect the channel to be initially unusable and waiting for a channel activation of TCH/F, TCH/H or PDCH; i.e. BSC shall send explicit CHAN ACT.

#4 - 07/13/2016 05:33 PM - neels

- Status changed from New to In Progress

#5 - 07/13/2016 05:38 PM - neels

neels wrote:

2.2 lchan_alloc(): pick and switch dyn channels (4h)

- TCH/H: pick second TCH/H if such TS is used only by half

To clarify, even if a dynamic TS is running in TCH/H mode and still has a free TCH/H channel, the picking algorithm should **always** favor a dedicated and available TCH/H channel over a dynamic one.

#6 - 07/14/2016 12:46 AM - laforge

- Related to Bug #1564: dynamic TCH/F + TCH/H allocation added

#7 - 07/14/2016 01:14 AM - laforge

On Wed, Jul 13, 2016 at 05:33:04PM +0000, neels [REDMINE] wrote:

- How does the ericsson BTS expect this?

The channel-activation message has been expanded to include a PDCH type.
So I expect the channel to be initially unusable and waiting for a channel
activation of TCH/F, TCH/H or PDCH; i.e. BSC shall send explicit CHAN ACT.

correct. Just like it is with any other classic circuit-switched
channel: The channel is unused for traffic (not unusable) until
activated. FYI: The otherwise channel might meanwhile be performing
uplink interference measurements and report them to the BTS, yet again
something we don't do yet (and are not implementing as part of this
ticket, for sure).

--

- Harald Welte <laforge@gnumonks.org> <http://laforge.gnumonks.org/>

=====
"Privacy in residential applications is a desirable marketing option."
(ETSI EN 300 175-7 Ch. A6)

#8 - 07/14/2016 01:18 PM - neels

- % Done changed from 0 to 10

commenced branches named neels/dyn_ts in libosmocore and openbsc.

#9 - 07/14/2016 11:32 PM - neels

- % Done changed from 10 to 30

#10 - 07/18/2016 09:50 AM - neels

- % Done changed from 30 to 50

Progress is looking good so far.

#11 - 07/21/2016 12:20 PM - neels

Currently stuck on a detail that one of two TCH/H is not initializing properly,
but should find that bug soon...

#12 - 07/21/2016 02:25 PM - neels

- % Done changed from 50 to 80

Feature tested as working on SysmoBTS for the first time.
litecell15 should work with trivial patch.

Now focusing on polishing up the patch.

#13 - 07/21/2016 02:39 PM - neels

- % Done changed from 80 to 70

#14 - 07/21/2016 04:51 PM - neels

- % Done changed from 70 to 80

Trivial patch for litecell15 confirmed working.

#15 - 07/21/2016 04:56 PM - neels

Related problem: [#1778](#)

With fully dynamic TS, we're implicitly allowing both TCH/H and TCH/F,
so if an MS asks for a TCH/F, it will get one before even considering a
TCH/H, even if that mismatches the MS peer.

So in a dynamic TS configuration where one phone favors a TCH/F and the
other favors a TCH/H, we will end up with:

DCC <0001> gsm_04_08.c:1649 Cannot patch through call with different channel types: local = TCH_F, remote = TCH_H

Until that is resolved, I will make the channel allocator actually skip the TCH/F option of TCH/F_TCH/H_PDCH dynamic slots. (This matches a specific user's expectation to be able to use dynamic TCH/H <-> PDCH, which this will then offer.)

We could also offer an explicit TCH/H_PDCH type (or some other config to disable TCH/F on dyn TS, for example).

#16 - 07/21/2016 04:56 PM - neels

- Related to Feature #1778: avoid mismatching TCH/F vs TCH/H pchan types added

#17 - 07/21/2016 05:03 PM - neels

remaining tasks:

- clean up the patch to be worthy for submission, handle review comments
- test and improve dynamic switchover error recovery
- resolve [#1778](#) with respect to dynamic TS

#18 - 07/24/2016 09:08 PM - neels

- % Done changed from 80 to 90

The commits have been prepared for submission for merge to master. After clean up/refactoring, I need to do a final testing round to verify before actually submitting the core code. Expecting no problems there.

The first preparatory and cosmetic patches have already been submitted.

#19 - 07/25/2016 12:25 PM - neels

dyn TS implementations for osmo-nitb and osmo-bts-sysmo and -litecell15 are complete, polished and tested, and waiting on branches named 'neels/dyn_ts' in openbsc.git and osmo-bts.git.

To not flood gerrit with too many patches, waiting for the first preparatory patches to be accepted.

Particularly, an osmo-bts commit depends on <https://gerrit.osmocom.org/569> in openbsc (same commit log summary), so that's a good point to split patch submission.

#20 - 07/25/2016 08:17 PM - neels

All commits are now submitted for review in gerrit.

(Due to a problem with gerrit, some commits were submitted twice in osmo-bts.git, so that the second submissions are actually empty patches. Look for a previous commit with the same log message instead.)

#21 - 07/26/2016 05:20 PM - neels

All osmo-bts commits have been merged. Most openbsc commits have been approved, but three still need a +1.

osmo-trx: dyn PDCH [#1565](#) has just been resolved for osmo-trx, which uses the same internal API as fully dynamic TS. So TCH/F_TCH/H_PDCH should need only minimal effort to also work on osmo-trx, once [#1565](#) is committed.

The remaining issue for TCH/F on dyn TS is [#1778](#) (dyn TCH/F is implemented and works well, but is disabled on purpose because of [#1778](#)).

#22 - 07/28/2016 02:44 PM - laforge

- Target version set to Dynamic TCH/H TCH/F PDCH

#23 - 07/29/2016 05:28 PM - laforge

all related patches have been merged to master meanwhile. What's pending is the documentation update. Let's keep the ticket open as reminder.

#24 - 08/01/2016 07:37 AM - laforge

- *Status changed from In Progress to Closed*

- *% Done changed from 90 to 100*

#25 - 08/10/2016 05:34 PM - neels

- *Related to Bug #1796: PTCCH activation breaks dyn TS added*