SBA blocks allocated (reserved) are stored in SBAController class under gprs_rlcmac_bts object [bts_sba(bts)]. It sets the SBA block on FN + "AGCH_START_OFFSET=52" (also theoretically linked to timer X2002).

SBAController doesn't seem to have into account that a TBF may have already reserved that FN when allocating a SBA (hence either there may be a collision at that time).

Current allocation of TBF polls consist on always allocating poll on RRBPs as FN+13. This has the advantage that there's no need to check for collisions between different TBFs on a given PDCH, since on a given FN only a TBF is selected/executed and hence only that one can even reserve FN+13 (because the +13 is fixed offset). However, it may happen sometimes that a TBF tries to allocate FN+13 (tbf->check_polling()/set_polling()), but that FN+13 is already taken by a previous SBA which allocated it to FNb+52 such that FNb=FNa-52+13, and hence the TBF fails to allocate a poll for that CTRL message, which in the end it means it ends up failing to construct the CTRL message and the scheduler ends up sending a RLCMAC Dummy Block instead.

Since that can happen, the tbf->check_polling() should try harder finding a poll FN by trying other RRBP values than FN+13.

TBF class implementation of polling seems to only support 1 active poll at a time. That means, If for instance scheduler selects the same TBF at let's say FNa=10 and immediately after at FNb=10+4=14, If both where to request poll allocation, it would fail in tbf->check_polling() due to "poll_state != GPRS_RLCMAC_POLL_NONE" being hit. That's basically because the TBF class stores the active poll FN in class attribute "tbf->poll_fn/ts". So ideally there should be some sort of linkedlist or circular buffer or whatever supporting several active pollings. This becomes more important at the time we start using bigger values for RRBPs, where the "poll active" state can be there for longer periods and hence probability to hit this issue increments.

Related issues:
- Related to OsmoPCU - Bug #5033: N3101 potentially being updated only during R...
  - Resolved
  - 02/17/2021
- Related to OsmoPCU - Feature #5122: Choose SBA allocation offset based on AGC...
  - New
  - 04/20/2021

Revision 464627c0 - 03/05/2021 05:57 PM - pespin

pdch: Silently ignore DATA.ind with len=0

Since recently (see Depends below), BTS side submits DATA.ind with len=0 to announce nothing was received on that UL block FN. This will allow osmo-pcu track time more accurately, and use this information to quickly find out if a UL block was expected as requested by RRBPs or USF poll and increment counters such as N3101 (finally being able to properly implement timers such as T3619).

Depends: osmo-bts.git Change-Id I343c7a721db72411dedca816c8864926bc329fb
Related: OS#5020
Change-Id: l17c28lab63b153448b53971ac5c2e48daad4ea8

Revision 7a14d81b - 03/05/2021 06:00 PM - pespin

Track TDMA clock with DATA.ind instead of TIME.ind
Since recently (see Depends below), BTS side submits DATA.ind with len=0 to announce nothing was received on that UL block FN. This will allow osmo-pcu track time more accurately, and use this information to quickly find out if a UL block was expected as requested by RRBP or USF poll and increment counters such as N3101 (finally being able to properly implement timers such as T3619).

Depends: osmo-bts.git Change-Id I343c7a7211dab72411edbca816c8864926bc329fb
Related: OS#5020
Change-Id: lbc495173119465e74f726ddc36e312334e6dc0fd

Revision 60ed844b - 03/05/2021 06:15 PM - pespin
pcu: transmit PCUIF DATA.ind with len=0 when no UL data to transmit

PCUIF will be updated to always send DATA.ind for each expected block FN on any activated PDCH slot, irrespective of whether valid data was received or not, similarly to what's done already for TRXDv1 NOPE.ind in TRXD and TCH channels in OsmoBTS. The aim at this change is to be able to track TDMA clock in an accurate way without hops, and hence be able to detect on time whether expected UL blocks (SF, RRBP poll) didn't arrive.

Older osmo-pcu versions can cope well with this change, they will simply print an error upon ach data_len=0 messages received and submit a GSMTAP block, then discard it, so tests still pass.

Nevertheless, a new module parameter is added to disable this new behavior in order to avoid logs and pcap files ending up clogged with unneeded information until a new osmo-pcu release appears.

Related: OS#5020
Change-Id: lb4f97a9bca68230945effeb6412218faa64ec78

Revision 70b3c3ca - 03/08/2021 09:33 AM - pespin
ttcn3-pcu: Disable sending all DATA.ind on pcu -latest

Depends: osmo-ttcn3-hacks.git Change-Id lb4f97a9bca68230945effeb6412218faa64ec78
Related: OS#5020
Change-Id: ld265d08a316bc803c565c3ca465bc19f1088b92

Revision 3c64d7 - 03/08/2021 09:39 AM - pespin
pdch: Silently ignore DATA.ind with len=0

Since recently (see Depends below), BTS side submits DATA.ind with len=0 to announce nothing was received on that UL block FN. This will allow osmo-pcu track time more accurately, and use this information to quickly find out if a UL block was expected as requested by RRBP or USF poll and increment counters such as N3101 (finally being able to properly implement timers such as T3619).

Depends: osmo-bts.git Change-Id I343c7a7211dab72411edbca816c8864926bc329fb
Related: OS#5020
Change-Id: l17c28af63b153448b533971ac5c2e48daa6e8

Revision 1fb91ddc - 03/08/2021 12:11 PM - pespin
Track TDMA clock with DATA.ind instead of TIME.ind

Since recently (see Depends below), BTS side submits DATA.ind with len=0 to announce nothing was received on that UL block FN. This will allow osmo-pcu track time more accurately, and use this information to quickly find out if a UL block was expected as requested by RRBP or USF poll and increment counters such as N3101 (finally being able to properly implement timers such as T3619).

Depends: osmo-bts.git Change-Id I343c7a7211dab72411edbca816c8864926bc329fb
Related: OS#5020
Change-Id: lbc495173119465e74f726ddc36e312334e6dc0fd

Revision 6531614d - 03/08/2021 07:59 PM - pespin
WIP: Add new PDCH UL Controller, drop SBAllocator class
Right now we handle different types of UL allocations in different classes like PollAllocator and SBAlocator, and they usually don't take into account the other one in most cases. Furthermore, those objects are usually per-BTS object, instead of per PDCH object.

This is a first step towards having a unified per-PDCH controller which takes care of controlling what is scheduled and hence expected on the uplink. Each PDCH has a UL Controller which keeps track of all reserved uplink frame, be it SB, RRBP poll or USF assigned, all under the same API.

As a first step, only the SBA part is fully implemented and used (being it the easiest part to replace); TBF poll+usf will come in follow-up patches later on. As a result, the SBAlocator per-BTS class dissappears but some of its code is refactored/reused to provide more features to the gprs_rlcmac_sba object, which is also further integrated into the new UL Controller.

Related: OS#5020
Change-Id: I84b24beea4a1aa2c1528f414135f77bd16df2b947

**Revision e130af6 - 03/09/2021 07:20 PM - pespin**

WIP: Add new PDCH UL Controller, drop SBAlocator class

Right now we handle different types of UL allocations in different classes like PollAllocator and SBAlocator, and they usually don't take into account the other one in most cases. Furthermore, those objects are usually per-BTS object, instead of per PDCH object.

This is a first step towards having a unified per-PDCH controller which takes care of controlling what is scheduled and hence expected on the uplink. Each PDCH has a UL Controller which keeps track of all reserved uplink frame, be it SB, RRBP poll or USF assigned, all under the same API.

As a first step, only the SBA part is fully implemented and used (being it the easiest part to replace); TBF poll+usf will come in follow-up patches later on. As a result, the SBAlocator per-BTS class dissappears but some of its code is refactored/reused to provide more features to the gprs_rlcmac_sba object, which is also further integrated into the new UL Controller.

Related: OS#5020
Change-Id: I84b24beea4a1aa2c1528f414135f77bd16df2b947

**Revision be315985 - 03/10/2021 11:22 AM - pespin**

Add new PDCH UL Controller, drop SBAlocator class

Right now we handle different types of UL allocations in different classes like PollAllocator and SBAlocator, and they usually don't take into account the other one in most cases. Furthermore, those objects are usually per-BTS object, instead of per PDCH object.

This is a first step towards having a unified per-PDCH controller which takes care of controlling what is scheduled and hence expected on the uplink. Each PDCH has a UL Controller which keeps track of all reserved uplink frame, be it SB, RRBP poll or USF assigned, all under the same API.

As a first step, only the SBA part is fully implemented and used (being it the easiest part to replace); TBF poll+usf will come in follow-up patches later on. As a result, the SBAlocator per-BTS class dissappears but some of its code is refactored/reused to provide more features to the gprs_rlcmac_sba object, which is also further integrated into the new UL Controller.

Related: OS#5020
Change-Id: I84b24beea4a1aa2c1528f414135f77bd16df2b947

**Revision e1142cf6 - 03/10/2021 11:22 AM - pespin**

Replace PollController with newly added PDCH UL Controller
TbfTest is updated to submit empty blocks to have somehow meaningful output (at least as meaningful test results as before, not much). That's because we must update bts->curr_fn to have polls expire.

Related: OS#5020
Change-Id: l683ca738ce5a133c49c36a1d94439a942d64a831

Revision 517d9cc8 - 03/10/2021 02:50 PM - pespin

Replace PollController with newly added PDCH UL Controller

TbfTest is updated to submit empty blocks to have somehow meaningful output (at least as meaningful test results as before, not much). That's because we must update bts->curr_fn to have polls expire.

Related: OS#5020
Change-Id: l683ca738ce5a133c49c36a1d94439a942d64a831

Revision 6a1a5f98 - 03/11/2021 05:41 PM - pespin

l1sap: Transmit pdtch invalid MAC blocks to PCU

Similar to what we have been doing for TCH channels, we want to make sure all MAC blocks get to the upper layers, even if containing invalid data (flagging it with data_len=0) so that upper layers (osmo-pcu through PCUIF in this case) can rely on FN clock without gaps due to Rx errors.

Related: OS#5020
Change-Id: l0b04b013b7bad5ff53d6a969ff0128b37f7f62d5

Revision 166b1005 - 03/11/2021 05:41 PM - pespin

bts-trx: Always submit rx PDTCH DATA.ind to l1sap

Similar to what we have been doing for TCH channels, we want to make sure all MAC blocks get to the upper layers, even if containing invalid data (flagging it with data_len=0) so that upper layers (osmo-pcu through PCUIF in this case) can rely on FN clock without gaps due to Rx errors.

Related: OS#5020
Change-Id: l343c7a721dab72411edbca816c8864926bc329fb

Revision 7a7beb9b - 03/11/2021 05:42 PM - pespin

bts-trx: Avoid submitting first data_ind with FN=0 to upper layers

It can happen that the first burst we receive after enabling the PDCH channel (when PCU connects to the BTS) is bid!=0. As a result, chan_state->ul_first_fn is never set and default value 0 in there is passed to the upper layers. As a result, when the 2nd block is transmitted, this time with correct FN, the PCU will see a huge jump in FNs. Since in PDCH the bursts are always consecutive, let’s simply use bi->fn - 3 as a first_fn and be done with the issue.

Related: OS#5020
Change-Id: le982caeb29f3fd880b44e88a89b85ea3e6e6947

Revision 0a27ded6 - 03/11/2021 06:36 PM - pespin

Add new PDCH UL Controller, drop SBAllocator class

Right now we handle different types of UL allocations in different classes like Poli Allocator and SBAllocator, and they usually don’t take into account the other one in most cases. Furthermore, those objects are usually per-BTS object, instead of per PDCH object.

This is a first step towards having a unified per-PDCH controller which takes care of controlling what is scheduled and hence expected on the uplink. Each PDCH has a UL Controller which keeps track of all reserved uplink frame, be it SB, RRBP poll or USF assigned, all under the same API.

As a first step, only the SBA part is fully implemented and used (being
it the easiest part to replace); TBF poll+usf will come in follow-up patches later on. As a result, the SBAllocat per-BTS class dissappears but some of its code is refactored/reused to provide more features to the gprs_rlcmac_sba object, which is also further integrated into the new UL Controller.

Related: OS#5020
Change-Id: l84b24beea4a1aa2c1528f41435f77bd16df2b947

Revision 6ef23910 - 03/11/2021 06:36 PM - pespin
Replace PollController with newly added PDCH UL Controller
TbfTest is updated to submit empty blocks to have somehow meaningful output (at least as meaningful test results as before, not much). That's because we must update bts->curr_fn to have polls expire.

Related: OS#5020
Change-Id: l683ca738cae5a133c49c36a1d94439a942d64a831

Revision b671c50c - 03/11/2021 06:36 PM - pespin
sched: Use new PDCH UL Controller
Take the time to also do small refactorings to clarify and simplify the function, by using rts_next_fn() already available in pcu_utils.h and getting rid of poll_tbf from tbf_candidates, which clearly follows another objective.

Using PDCH UL Controller has the advantage that we don't need to check poll_scheduled() on each TBF, but only do the query once.

Related: OS#5020
Change-Id: la60bb5249a9837dec1f4218044d984833486d6

Revision 30617115 - 03/12/2021 07:40 AM - pespin
Track TDMA clock with DATA.ind instead of TIME.ind
Since recently (see Depends below), BTS side submits DATA.ind with len=0 to announce nothing was received on that UL block FN. This will allow osmo-pcu track time more accurately, and use this information to quickly find out if a UL block was expected by RBBP or USF poll and increment counters such as N3101 (finally being able to properly implement timers such as T3619).

Depends: osmo-bts.git Change-Id l343c7a6721da72411edbca816c88649266cc329fb
Related: OS#5020
Change-Id: lbc495173119465e74f726ddc36e3123346dc0fd

Revision 873686f0 - 03/12/2021 11:44 AM - pespin
pcu: transmit PCUIF DATA.ind with len=0 when no UL data to transmit
PCUIF will be updated to always send DATA.ind for each expected block FN on any activated PDCH slot, irrespective of whether valid data was received or not, similarly to what's done already for TRXDV1 NOPE.ind in TRXD and TOH channels in OsmoBTS. The aim at this change is to be able to track TDMA clock in an accurate way without hops, and hence be able to detect on time whether expected UL blocks (SF, RBBP poll) didn't arrive.

Older osmo-pcu versions can cope well with this change, they will simply print an error upon ach data_len=0 messages received and submit a GSMTAP block, then discard it, so tests still pass. Nevertheless, a new module parameter is added to disable this new behavior in order to avoid logs and pcap files ending up clogged with unneeded information until a new osmo-pcu release appears.

Related: OS#5020
Change-Id: lbf497a9bfb68230945eefeb6412218faa64ec78

Revision a8b0dbce - 03/12/2021 11:49 AM - pespin
pcu: Set up PCU TDMA clock by sending initial DATA.ind

05/13/2021
In recent osmo-pcu commits, initial fn was changed to invalid value -1, in order to be able to detect FN jumps. Previously, the initial value was set randomly to 0, which was wrong anyway because first FN received from the BTS could be any other FN counted by the BTS at that time.

This makes some tests fail because they send RACH.ind + RTS.ind to receive the Imm Assignment, and the Request reference in the Imm Assign was calculated on the invalid unset FN "-1", hence it won't match test expectancies.

In order to fix it, simply make sure the TDMA clock is initiated by sending a DATA.ind to the PCU before tests start doing stuff (f_init_raw() is blocked waiting for BTS_EV_SI13_NEGO).

Related: osmo-pcu.git Change-Id l29fb27981597dc69abb976049ba41aa840488cb
Related: OS#5020
Change-Id: l00c4dd9133ec9a236bf28fb8cb0a0d0615791012

Revision f8d5d500 - 03/12/2021 11:51 AM - pespin

ttc3-pcu: Disable sending all DATA.ind on pcu -latest

Change-Id: l4365d54c64e750a708e04a36ea131ec7499560f1
Depends: osmo-ttcn3-hacks.git Change-Id lb4f97a9bca68230945e6feeb6412218f9a64ec78
Related: OS#5020

Revision ed9ca971 - 03/12/2021 11:59 AM - pespin

pcu: Set up PCU TDMA clock by sending initial DATA.ind

In recent osmo-pcu commits, initial fn was changed to invalid value -1, in order to be able to detect FN jumps. Previously, the initial value was set randomly to 0, which was wrong anyway because first FN received from the BTS could be any other FN counted by the BTS at that time.

This makes some tests fail because they send RACH.ind + RTS.ind to receive the Imm Assignment, and the Request reference in the Imm Assign was calculated on the invalid unset FN "-1", hence it won't match test expectancies.

In order to fix it, simply make sure the TDMA clock is initiated by sending a DATA.ind to the PCU before tests start doing stuff (f_init_raw() is blocked waiting for BTS_EV_SI13_NEGO).

Related: osmo-pcu.git Change-Id l29fb27981597dc69abb976049ba41aa840488cb
Related: OS#5020
Change-Id: l00c4dd9133ec9a236bf28fb8cb0a0d0615791012

Revision 68b4d3fe - 03/12/2021 01:58 PM - pespin

Add new PDCH UL Controller, drop SBAlocator class

Right now we handle different types of UL allocations in different classes like PollAllocator and SBAlocator, and they usually don't take into account the other one in most cases. Furthermore, those objects are usually per-BTS object, instead of per PDCH object.

This is a first step towards having a unified per-PDCH controller which takes care of controlling what is scheduled and hence expected on the uplink. Each PDCH has a UL Controller which keeps track of all reserved uplink frame, be it SB, RRBP poll or USF assigned, all under the same API.

As a first step, only the SBA part is fully implemented and used (being it the easiest part to replace); TBF poll+usf will come in follow-up patches later on. As a result, the SBAlocator per-BTS class dissappears but some of its code is refactored/reused to provide more features to the gprs_rfcmac_sba object, which is also further integrated into the new UL Controller.

Related: OS#5020
Change-Id: l84b24beea4a1aa2c152841435f77bd16df2b947

Revision 4eae1411 - 03/12/2021 01:58 PM - pespin
Replace PollController with newly added PDCH UL Controller

TbfTest is updated to submit empty blocks to have somehow meaningful output (at least as meaningful test results as before, not much). That’s because we must update bts->curr_fn to have polls expire.

Related: OS#5020
Change-Id: I683ca738ce5a133c49c36a1d94439a942d64a831

Revision 7bdbb8b22 - 03/12/2021 02:05 PM - pespin
sched: Use new PDCH UL Controller

Take the time to also do small refactorings to clarify and simplify the function, by using rts_next_fn() already available in pcu_utils.h and getting rid of poll_tbf from tbf_candidates, which clearly follows another objective.

Using PDCH UL Controller has the advantage that we don’t need to check poll_scheduled() on each TBF, but only do the query once.

Related: OS#5020
Change-Id: la60bb5249a9837dec1f42180e44d9848334d86d6

Revision b6e746d5 - 03/12/2021 03:56 PM - pespin
tests: Introduce unit tests for PDCH UL Controller

Related: OS#5020
Change-Id: le1ff0ca3d7fc8a9824d6fe4dceb746e301082bda

Revision 0747c127 - 03/12/2021 04:50 PM - pespin
tests: ulc: Show current bug with FN wrap around

Issue will be fixed in next commit. Leaving ASSERTs disabled so that test passes in jenkins.

Related: OS#5020
Change-Id: i657db6b300363f8f3a9e4cafa7a7f49e361a0512

Revision 3b05e324 - 03/12/2021 05:25 PM - pespin
ulc: Fix FN store order upon wrap around

Related: OS#5020
Change-Id: l0a7427fa1541b1837739207b9383772981fb25

Revision 15c58ace - 03/15/2021 06:32 PM - pespin
Add new PDCH UL Controller, drop SBAlocator class

Right now we handle different types of UL allocations in different classes like PollAllocator and SBAlocator, and they usually don't take into account the other one in most cases. Furthermore, those objects are usually per-BTS object, instead of per PDCH object.

This is a first step towards having a unified per-PDCH controller which takes care of controlling what is scheduled and hence expected on the uplink. Each PDCH has a UL Controller which keeps track of all reserved uplink frame, be it SB, RRBP poll or USF assigned, all under the same API.

As a first step, only the SBA part is fully implemented and used (being it the easiest part to replace); TBF poll+usf will come in follow-up patches later on. As a result, the SBAlocator per-BTS class disappears but some of its code is refactored/reused to provide more features to the gprs_rlcmac_sba object, which is also further integrated into the new UL Controller.

Related: OS#5020
Change-Id: l84b24beaa4a1aa2c152841435f77bd16df2b947

Revision 99360a30 - 03/15/2021 06:32 PM - pespin
Replace PollController with newly added PDCH UL Controller

TbfTest is updated to submit empty blocks to have somehow meaningful output (at least as meaningful test results as before, not much). That's because we must update bts->curr_fn to have polls expire.

Related: OS#5020
Change-Id: i683ca738ce5a133c49c36a1d94439a942d64a831

Revision fd1fbd8 - 03/15/2021 06:34 PM - pespin
sched: Use new PDCH UL Controller

Take the time to also do small refactorings to clarify and simplify the function, by using rts_next_fn() already available in pcu_utils.h and getting rid of poll_tbf from tbf_candidates, which clearly follows another objective.

Using PDCH UL Controller has the advantage that we don't need to check poll_scheduled() on each TBF, but only do the query once.

Related: OS#5020
Change-Id: la60bb5249a9837dec14f2180e44d9848334d86d6

Revision 582a15e4 - 03/15/2021 06:35 PM - pespin
tests: Introduce unit tests for PDCH UL Controller

Related: OS#5020
Change-Id: le1ff0ca3d7fc8a9824d6fe4dceb746e301082bda

Revision 95f8fa1f - 03/15/2021 06:36 PM - pespin
tests: ulc: Show current bug with FN wrap around

Issue will be fixed in next commit. Leaving ASSERTs disabled so that test passes in jenkins.

Related: OS#5020
Change-Id: i657db6b300363f8f3a9e4cfa7a7149e361a0512

Revision c7cc4162 - 03/15/2021 06:36 PM - pespin
ulc: Fix FN store order upon wrap around

Related: OS#5020
Change-Id: l0a742f7fa1541b1837739207b9383772f981f2b25

Revision 755a8d61 - 03/18/2021 01:03 PM - pespin
direct_phy: Fix condition dropping rx DATA.ind payload in in

Related: OS#5020
Fixes: 81c549d5be1f5e161d6231d3f2e5fb4aa3b0557c
Change-Id: lad8e50b856009439d78c596c5b54dc3e98361e1d4

Revision 107e94c9 - 03/24/2021 04:12 PM - pespin
sched: Fix scheduling UL TBF not matching conditions

With previous code, a skipped TBF could be returned despite not matching the conditions, since at the end of the loop the tbf pointer was returned.

Related: OS#5020
Change-Id: lf6dcecc86c7a655bf1c62f333cfbc8d2c507c94f

Revision 8238739a - 03/29/2021 11:54 AM - pespin
sba: Document AGCH_START_OFFSET after some experimental tests

Related: OS#5020
Change-Id: ld1460207be25750aeb5c1d7a2fac6591cf5e424

Revision 2f59e673 - 03/29/2021 11:55 AM - pespin
Pick unreserved UL FN when allocating an SBA

Make sure an unreserved FN is picked and reserved when allocating and scheduling an SBA.
In practice this has no change in behavior right now, since anyway using an offset of 52 FNs ensure no USF or POLL has already been scheduled that far in the future. Since it's also impossible to allocate more than 1 SBA per PDCH and RTS FN, we are also safe about multiple SBAs being allocated, because we use a hardcoded offset of 52.
However, that could change in the future, when we dynamically tweak the current offset of 52 FN based on information from BTS about its AGCH queue load:

- If load is high, we may need to increase the offset since it will take more time for the BTS to transmit the TBF and hence we must reserve a TBF starting time further in the future (higher FN).
- If load turns low, we may schedule next SBA a bit more nearby in time than the previously allocated SBA, hence here there could be a collision.

Related: OS#5020
Change-Id: l2d4e21e2307de6c8da5e7e149c947a7eb9

Revision 4eb16eecc - 03/29/2021 11:55 AM - pespin
pdch_ulc: Support picking RRBP other than N+13

Current algo always tries to sched RRBP the soonest possible.

Related: OS#5020
Change-Id: lc6dddea70e1f914cf423d0daab8fc492dc992e2

Revision 75a862d1 - 03/29/2021 02:04 PM - pespin
pdch_ulc: Support picking RRBP other than N+13

Current algo always tries to sched RRBP the soonest possible.

Related: OS#5020
Change-Id: lc6dddea70e1f914cf423d0daab8fc492dc992e2

Revision 1b5037bb - 03/29/2021 04:16 PM - pespin
pdch_ulc: Store TBF poll reason
This allows easily checking the initial reason to trigger the poll when either it is received or times out.

Later on this reason can be transformed into an FSM event and sent to the related FSM.

Related: OS#5020
Change-Id: le8fefd1147ad674ce597a8065b1528408956bde

Revision 231e659d - 03/29/2021 04:19 PM - pespin
pdch_ulc: Store TBF poll reason
This allows easily checking the initial reason to trigger the poll when either it is received or times out.

Later on this reason can be transformed into an FSM event and sent to the related FSM.

Related: OS#5020
Change-Id: le8fefd1147ad674ce597a8065b1528408956bde

Revision 54e64502 - 03/31/2021 03:39 PM - pespin
sba: Document AGCH_START_OFFSET after some experimental tests

Related: OS#5020
Change-Id: ld1460207be25750abc5cfd7af2fac6591cf5e424

Revision ce3dd252 - 03/31/2021 03:39 PM - pespin
Pick unreserved UL FN when allocating an SBA

Make sure an unreserved FN is picked and reserved when allocating and scheduling an SBA.
In practice this has no change in behavior right now, since anyway using an offset of 52 FNs ensure no USF or POLL has already been scheduled that far in the future. Since it’s also impossible to allocate more than 1 SBA per PDCH and RTS FN, we are also safe about multiple SBAs being allocated, because we use a hardcoded offset of 52.
However, that could change in the future, when we dynamically tweak the current offset of 52 FN based on information from BTS about its AGCH queue load:

- If load is high, we may need to increase the offset since it will take more time for the BTS to transmit the TBF and hence we must reserve a TBF starting time further in the future (higher FN).
- If load turns low, we may schedule next SBA a bit more nearby in time than the previously allocated SBA, hence here there could be a collision.

Related: OS#5020
Change-Id: l2d4e21e2307de6c17748e8da5c7e149c947a7eb9

Revision 50a1ed6 - 03/31/2021 03:39 PM - pespin
pdch_ulc: Support picking RRBP other than N+13
Current algo always tries to sched RRBP the soonest possible.

Related: OS#5020
Change-Id: lc6ddeeaa70e1f914cf423d0daab8fc492d0c992e2

Revision 86580e19 - 03/31/2021 03:39 PM - pespin
pdch_ulc: Store TBF poll reason
This allows easily checking the initial reason to trigger the poll when either it is received or times out.
Later on this reason can be transformed into an FSM event and sent to the related FSM.
Related: OS#5020
Change-Id: le8feff1f47ad674ce597a8065b1528408956bde

History
#1 - 02/11/2021 05:33 PM - pespin

The last point (TBF only supporting 1 active poll FN) is worse than expected, because PollController seems to be timing out non-acked polls for a larger period of time than the FN+13 one:

```c
void bts_set_current_frame_number(struct gprs_rlcmac_bts *bts, int fn)
{
    /* The UL frame numbers lag 3 behind the DL frames and the data indication is only sent after all 4 frames of the block have been received. Sometimes there is an idle frame between the end of one and start of another frame (every 3 blocks). So the timeout should definitely be there if we're more than 8 frames past poll_fn. Let's stay on the safe side and say 13 or more. An additional delay can happen due to the block processing time in the DSP, so the delay of decoded blocks relative to the timing clock can be much larger. */
    const static int max_delay = 60;
    bts->cur_fn = fn;
    bts->pollController->expireTimedout(bts->cur_fn, max_delay);
}
```

For the ACKed one's is fine, since gprs_rlcmac_pdch::rcv_control_ack() calls:

```c
TBF_POLL_SCHED_UNSET(tbf);
```
Some more facts after further investigation:

- `bts->cur_fn` holds the current/last MAC block received on the UL, so it drives the clock.
- The `bts->cur_fn` field is updated in 2 code paths:
  - `bts_set_current_frame_number`: the main way, triggered by PCUIF time_ind received on every start of MAC block FN from BTS (even when direct PHY is used, TIME.IND in lower layers are only received in BTS)
  - `bts_set_current_block_frame_number`: used only by sysmo direct PHY (not used by other direct PHY like o2g) to correct the clock

So, the idea here would be to:
1. Make sure generic osmo-pcu code supports receiving `data_len=0` minimally: `"pcu_rx_data() => pcu_rx_data_ind_pdtch() => pdch->rcv_block()"
   This should already be fine OK since `mcs_get_by_size_ul()` will return `UNKNOWN=0` and early return printing an error. So this first step is done.
2. Make sure all phys always send `data_ind` even if no data received, or corrupted, by setting `data_len = 0` (aka NOPE.ind)
3. In osmo-pcu, ignore received time_ind messages (at most cross-check the FN and print warning otherwise), and update the clock (`bts_set_current_frame_number`) during `data_ind` instead.
4. In osmo-pcu upper layers, stored scheduled USF for that BTS+TRX+TS+FN and if `data_ind data_len=0` (NOPE.ind), then increment N3101 for that TBF.

Step 2 (backend sending NOPE.ind `data_len=0`) requires:
2.1- Allow direct PHY osmo-pcu code to submit corrupt/empty `data_ind`. For example, for sysmo direct PHY:

```c
handle_ph_data_ind (sysmo_l1_if.c)
pcu_rx_block_time
  bts_set_current_block_frame_number \--- RX FN number is corrected here
  bts->pollController->expireTimeout(fn, max_delay);
if (data_ind->msgUnitParam.u8Value == 0)
  return -1; \--- NOPE.ind is discarded here
  /* drop incomplete UL block */
  if (data_ind->msgUnitParam.u8Buffer[0] != GsmL1_PdtchPlType_Full) break;
  \--- NOPE.ind is discarded here
pcu_rx_data_ind_pdtch
```

2.2- Allow indirect PHY from osmo-bts to send PCUIF INFO_IND with `data_len=0`. By example, for sysmo direct PHY:

```c
l1sap_ph_data_ind:
if (ts_is_pdch(&trx->ts[tn])) {
  ...
  /* don't send bad frames to PCU */
  if (len == 0)
    return -EINVAL; \--- NOPE.ind is discarded here
   /* drop incomplete UL block */
  if (pr_info != PRES_INFO_BOTH)
    return 0; \--- NOPE.ind is discarded here
  pcu_tx_data_ind(...) \----- This function contains a memcpy(x, y, len) which must be done conditional on len>0 (undefined behavior of memcpy for len=0).
  return 0;
}
```

#4 - 03/05/2021 06:32 PM - pespin

I updated osmo-bts-trx, osmo-pcu and TTCN3 PCUIF_Components to send all DATA.ind even if `len=0`. Related patches:

osmo-bts.git:
- https://gerrit.osmocom.org/c/osmo-bts/+23249 l1sap: Transmit pdtch invalid MAC blocks to PCU
- https://gerrit.osmocom.org/c/osmo-bts/+23257 bts-trx: Always submit rx PDTCH DATA.ind to l1sap
- TODO: implement change for lower layers of other bts types!

osmo-pcu.git:
- https://gerrit.osmocom.org/c/osmo-pcu/+23260 pdch: Silently ignore DATA.ind with len=0
- https://gerrit.osmocom.org/c/osmo-pcu/+23261 Track TDMA clock with DATA.ind instead of TIME.ind

osmo-ttcn3-hacks.git:
- https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+23258 pcu: transmit PCUIF DATA.ind with len=0 when no UL data to transmit
**docker-plaground.git:**

- [https://gerrit.osmocom.org/c/docker-playground/+23262](https://gerrit.osmocom.org/c/docker-playground/+23262) ttcn3-pcu: Disable sending all DATA.ind on pcu -latest

I did some manual tests with osmo-bts-trx and osmo-pcu patches applied and everything looks good. I also checked that TTCN3 patches still work fine with older versions of osmo-pcu.

osmo-ttcn3-hacks + docker can already be merged. The first osmo-pcu patch silently ignoring data_len=0 too. Other ones I want to first implement it for all BTS types and are marked as WIP in gerrit.

### #5 - 03/08/2021 12:28 PM - pespin

pespin wrote:

```
osmo-bts.git:
  - TODO: implement change for lower layers of other bts types!
```

Actually the code in osmo-bts.git is OK and already letting pass of size=0 and alike, due to previous work we already did with TCH channels. The missing changes are in osmo-pcu for the direct phy case. I submitted a patch here:

[https://gerrit.osmocom.org/c/osmo-pcu/+23268](https://gerrit.osmocom.org/c/osmo-pcu/+23268)

```
direct_phy: Support submitting DATA.ind with len=0 to upper layers
```

### #6 - 03/24/2021 06:18 PM - pespin

- Status changed from New to In Progress
- % Done changed from 0 to 60

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23486](https://gerrit.osmocom.org/c/osmo-pcu/+23486)

Fix: left shift cannot be represented in type int [NEW]

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23487](https://gerrit.osmocom.org/c/osmo-pcu/+23487)

sched: Fix scheduling UL TBF not matching conditions [NEW]

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23488](https://gerrit.osmocom.org/c/osmo-pcu/+23488)

sched: Simplify usf selection code [NEW]

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23490](https://gerrit.osmocom.org/c/osmo-pcu/+23490)

pdcch: Add missing pdcch_ulc_release_node in Rx Cell Change NOTIF [NEW]

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23491](https://gerrit.osmocom.org/c/osmo-pcu/+23491)

pdcch_ulc: Create helper API pdcch_ulc_release_node [NEW]

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23492](https://gerrit.osmocom.org/c/osmo-pcu/+23492)

Track scheduled UL blocks through USF [NEW]

remote:
[https://gerrit.osmocom.org/c/osmo-pcu/+23493](https://gerrit.osmocom.org/c/osmo-pcu/+23493)

Properly implement N3101 [NEW]

After this bunch of patches become merged, we already track all sort of UL blocks in a unified way. What’s missing regarding this ticket:

- Improve SBA scheduler to offer something else than N+52 (AGCH_START_OFFSET)
- Improve SBA/POLL scheduler to offer something else than N+13

### #7 - 03/26/2021 01:07 PM - pespin

So, as per SBA allocation (Imm Assignment on CCCH):

The scheduled FN for the SBA is currently hardcoded to 52 (sba.c):

```c
/* starting time for assigning single slot */
#define AGCH_START_OFFSET 52

struct gprs_rlc_mac_sba *sba_alloc(void *ctx, struct gprs_rlc_mac_pdcch *pdcch, uint8_t ta)
{
    struct gprs_rlc_mac_sba *sba;
    sba = talloc_zero(ctx, struct gprs_rlc_mac_sba);
    if (!sba)
        return NULL;
    sba->pdch = pdcch;
    sba->ta  = ta;

    /* TODO: request ULC for next available FN instead of hardcoded AGCH_START_OFFSET */
    sba->fn  = next_fn(pdcch->last_rts_fn, AGCH_START_OFFSET);

    pdcch_ulc_reserve_sba(pdcch->ulc, sba);
    return sba;
}
```

Then, that FN is fed into Imm Assignment when encoding it (bts.cpp bts_rcv_rach):
sba = bts_alloc_sba(bts, ta);
sb_fn = sba->fn;
plen = Encoding::write Immediate_assignment(
 &bts->trx[trx_no].pdch[ts_no], tbf, bv,
 false, rip->ra, Fn, ta, usf, false, sb_fn,
 bts_get_ms_pwr_alpha(bts), bts->pcu->vty.gamma, -1,
 rip->burst_type);

And sba_Fn is used here in Encoding::write_immediate_assignment as ref_fn to indicate the offset to the MS to start using it:

bitvec_write_field(dest, &wp, (ref_fn / (26 * 51)) % 32,5); // T1'
bitvec_write_field(dest, &wp, ref_fn % 51,6); // T3
bitvec_write_field(dest, &wp, ref_fn % 26,5); // T2

Now, the question is why is 52 being used, and why it must be multiple of 13... Need to find that in the spec.

The symbol AGCH_START_OFFSET and the related comment about being multiple of 13 was added here, without much explanation:

commit 07e97cf8a55b05d7f5f3f9583b68b2eff0f1c23
Author: Andreas Eversberg <jolly@eversberg.eu>
Date: Tue Aug 7 16:00:56 2012 +0200

Adding single block allocation

It is mandatory to support it because MS may request a single block.
In this case the network must assign a single block.

It is possible to force single block allocation for all uplink requests on RACH. (VTY option)

#8 - 03/29/2021 12:12 PM - pespin

After these patches, FN for SBA and RRBP poll is scheduled based on available/unreserved FNs in the UL scheduler:
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23520 Pick unreserved UL FN when allocating an SBA
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23522 pdch_ulc: Support picking RRBP other than N+13 [NEW]

So, after those are merged, what's TODO:

- modify SBA allocator offset in sba_alloc() based on AGCH queue load in the BTS
- Update gprs_rlcmac_tbf poll_state/poll_fn/poll_ts implementation to support multiple concurrent polls allocated.

#9 - 03/29/2021 05:32 PM - pespin
- % Done changed from 60 to 80

I submitted a bunch of commits to get rid of old tbf's poll_fn/poll_ts infra which only supported 1 concurrent poll request per tbf here:
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23525 pdch_ulc: Store TBF poll reason [NEW]
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23527 tbf: Get rid of unneeded poll_scheduled() [NEW]
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23528 tbf: Allow multiple concurrent polls [NEW]
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23529 Remove unneeded poll_state check [NEW]
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23530 tbf: Get rid of poll_state completely [NEW]
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23531 tbf: Get rid of attribute poll_fn [NEW]
remote: https://gerrit.osmocom.org/c/osmo-pcu/+/23532 tbf: Get rid of attribute poll_ts [NEW]

#10 - 04/20/2021 12:52 PM - pespin
- Related to Feature #5122: Choose SBA allocation offset based on AGCH queue load in the BTS added

#11 - 04/20/2021 12:53 PM - pespin
- Status changed from In Progress to Resolved
- % Done changed from 80 to 100

Patches were merged.
I created a separate ticket to track the SBA allocation fixed delay here:
https://osmocom.org/issues/5122
Hence, closing this ticket. Other detailed ticket can be created in the future to track specific issues/improvements.