OsmoPCU - Bug #5209

Memory usage on current master 945be910

08/06/2021 07:44 PM - iedemam

Status:	Feedback	**Start date:**	08/06/2021
Priority:	High	**Due date:**	
Assignee:	iedemam	**% Done:**	50%
Category:			
Target version:			
Spec Reference:			

Description

Hi all,

On a busy site with 4 TRX but only 4 PDCHs configured, we are seeing ever-increasing memory usage in OsmoPCU. This will continue until the system is exhausted of memory and processes are killed for OOM.

I'm attaching the PCU's "talloc-context application brief" report to show the state as we hit 1GB of RAM.

I wish I had more information on this one but will start a ticket to kick off the conversation. Let me know what would be useful.

Thanks,
-Michael

Associated revisions

Revision 41633619 - 08/12/2021 01:08 PM - Neels Hofmeyr

T_defs_bts: remove unit from doc strings

The main reason to change this is that the unit for T3172 is wrong. It is defined as ms but the doc string says "(s)".

The tdef implementation already includes the unit as defined for each T in the doc string implicitly, so instead of fixing that string, just remove the unit strings from all the doc strings.

Now it will show:

```
OsmoPCU# show bts-timer
 BTS0:
  T3142 = 20 s  Wait Indication used in Imm Ass Reject during TBF Establishment (CCCH) (default: 20 s, range:
[0 .. 255])
  T3169 = 5 s   Reuse of USF and TFI(s) after the MS uplink TBF assignment is invalid (default: 5 s)
  T3172 = 5000 ms       Wait Indication used in Imm Ass Reject during TBF Establishment (PACCH) (default: 5000
 ms, range: [0 .. 255000])
  T3191 = 5 s   Reuse of TFI(s) after sending (1) last RLC Data Block on TBF(s), or (2) PACKET TBF RELEASE for
 an MBMS radio bearer (default: 5 s)
  T3193 = 1600 ms       Reuse of TFI(s) after reception of final PACKET DOWNLINK ACK/NACK from MS for TBF (def
ault: 100 ms)
  T3195 = 5 s   Reuse of TFI(s) upon no response from the MS (radio failure or cell change) for TBF/MBMS radio
 bearer (default: 5 s)
```

Related: OS#5209
Change-Id: I140122bb10f750bf996272cc7f9c5b541c9bd364

Revision 112c63e9 - 08/15/2021 05:46 PM - Neels Hofmeyr

Revert "Stop abusing T3169"

This reverts commit 846fd248dc49c06441da6d7c3cd85df479810f1a.

The commit introduced a leak of UL-TBF, which do not time out and accumulate indefinitely, leading to out-of-memory for the running osmo-pcu process.

A proper fix for the leak is pending on a development branch pespin/fsm, but that branch is not yet ready for merging. Hence let's re-introduce

timer T3169 to avoid the OOM due to lingering UL-TBF.

Related: OS#5209
Change-Id: I99a7d2ddf68a76739ce2db1d6a44967dd97667b0

History

#1 - 08/07/2021 06:33 AM - laforge

ok, so it clearly shows that the uplink TBF is leaking, as we shouldn't have 6860 of those

#2 - 08/07/2021 06:50 AM - laforge

I've looked at the git log back to march 2021 and I couldn't immediately find something that might have affected ul_tbf memory release.

What was the last osmo-pcu version that you were using?

Completely random wild guess (my pcu knowledge is limited): Is your TRX implementing TRXDv1 or v2 fully? Not that the lack of indications for some frame numbers is creating downstream problems by code not being executed...

#3 - 08/07/2021 07:08 AM - laforge

some notes:

UL tbf **allocation** happens via (inverse call chain):

- tbf_ul.cpp: tbf_alloc_ul_tbf()
  - tbf_ul.cpp: tbf_alloc_ul_pacch()
    - pdch.cpp: sched_ul_ass_or_rej()
    - pdch.cpp: gprs_rlcmac_pdch::rcv_resource_request()
  - tbf_ul.cpp: tbf_alloc_ul_ccch()
    - bts.cpp: bts_rcv_rach()

UL tbf **free** happens from:

- gprs_ms.c: ms_merge_and_clear_ms()
  - when we discover we already have a MS object for the given MS
- pcu_if.c: pcu_rx_susp_req()
  - when we receive a suspend request from the CS domain
- pdch.cpp: pdch_free_all_tbf()
  - pdch.cpp: gprs_rlcmac_pdch::free_resources()
    - pdch.cpp: gprs_rlcmac_pdch::disable()
      - pcu_rx_info_ind() - when the BSC has disabled a PDCH
  - bts.cpp: bts_trx_free_all_tbf()
    - osmobts_sock.c: pcu_sock_close()
- pdch.cpp: gprs_rlcmac_pdch::rcv_control_ack()
  - after receiving the last CTRL ACK for a UL TBF
- pdch.cpp: gprs_rlcmac_pdch::rcv_resource_request()
  - when a new UL TBF is established for that MS, drop the old one

So what I can see from all of this is that

- most of the free paths are related to deactivation of PDCH/TRX/... which only happens in dynamic timeslots
- there is no guard timer or anything like that which would make a TBF time out
- normal free seems to happen via receiving the last CTRL ACK for that TBF
- I currently don't see the situation that somehow the MS is gone and we never receive that last CTRL ACK.

Keep in mind: I have not looked at the PCU code in a very long time, just wanted to give it a brief initial fresh set of eyes.

#4 - 08/07/2021 07:16 AM - laforge

@iedemam are you seeing any log messages about invalid state transitions? We introduced osmo_fsm in osmo-pcu in May and maybe some state change doesn't happen as expected, causing code paths not to be reached? In this case you should see "transition to state %s not permitted!\n", for TBF related objects.

#5 - 08/07/2021 07:23 AM - laforge

laforge wrote:

- there is no guard timer or anything like that which would make a TBF time out

I missed that gprs_rlcmac_ul_tbf is derived from gprs_rlcmac_tbf, and the latter has plenty of timeout handling, and every timeout is leading to a free.
So neither T3141, nor T3169, T3191, T3193, T3195 could have been running, as a timeout of any of those would lead to an immediate tbf_free().

#6 - 08/12/2021 12:32 PM - neels

- File osmo-pcu_TC_mt_ping_pong.log added

- File _PCU_Tests.TC_mt_ping_pong.merged.gz added

- Status changed from New to In Progress

- % Done changed from 0 to 50

Status update on how far I got with this issue:

I've added a check to the end of (almost) all TTCN3 PCU_Tests,
which counts the DL and UL TBF still allocated by fetching the talloc report via VTY.
If nonzero, it waits 5 seconds and checks again, repeats 10 times.

From this, I can identify a number of tests that still have UL TBF assigned when the test exits,
which don't seem to ever time out:

_PCU_Tests.TC_dl_egprs_data_no_llc_ui_dummy.merged
_PCU_Tests.TC_dl_gprs_data_no_llc_ui_dummy.merged
_PCU_Tests.TC_dl_multislot_tbf_ms_class_from_2phase.merged
_PCU_Tests.TC_dl_no_ack_retrans_imm_ass.merged
_PCU_Tests.TC_force_two_phase_access.merged
_PCU_Tests.TC_mo_ping_pong_with_ul_racap_egprs_only.merged
_PCU_Tests.TC_mo_ping_pong_with_ul_racap.merged
_PCU_Tests.TC_mt_ping_pong.merged
_PCU_Tests.TC_mt_ping_pong_with_dl_racap.merged
_PCU_Tests.TC_multiplex_dl_gprs_egprs.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_dup2.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_dup.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_twice2.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_twice3.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_twice4.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_twice5.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_twice.merged
_PCU_Tests.TC_nacc_outbound_pkt_cell_chg_notif_unassigned_dl_tbf.merged
_PCU_Tests.TC_nacc_outbound_rac_ci_resolve_conn_refused.merged
_PCU_Tests.TC_nacc_outbound_rac_ci_resolve_fail_parse_response.merged
_PCU_Tests.TC_nacc_outbound_success.merged
_PCU_Tests.TC_nacc_outbound_success_no_ctrl_ack.merged
_PCU_Tests.TC_nacc_outbound_success_twice.merged
_PCU_Tests.TC_nacc_outbound_success_twice_nocache.merged
_PCU_Tests.TC_paging_cs_multi_ms_imsi.merged
_PCU_Tests.TC_paging_cs_multi_ms_imsi_tmsi.merged
_PCU_Tests.TC_paging_cs_multi_ms_tmsi.merged
_PCU_Tests.TC_pcuif_fh_imm_ass_dl.merged
_PCU_Tests.TC_pcuif_fh_pkt_ass_dl.merged
_PCU_Tests.TC_pcuif_fh_pkt_ass_ul.merged
_PCU_Tests.TC_ul_multislot_tbf_ms_class_from_2phase.merged

I am now focusing on TC_mt_ping_pong and looking at its TBFs.
(There is a DL TBF, which gets deallocated after T3193.)
And there is an UL TBF which apparently never gets any timer set, and lingers after data is apparently transferred.

There is an osmo_fsm for the UL TBF which stays in state "ASSIGN".
There are no state timeouts defined on the osmo_fsm instances for TBF,
instead there is a list of timers gprs_rlcmac_tbf.Tarr[]

At the moment my main problem is lack of knowledge about how it **should** be timed out.
My usual approach would be to add a sanity timeout to the ASSIGN state of the TBF osmo_fsm instance.
Here though it seems that the design intends to not use any timeouts in the FSM.

Attaching the osmo-pcu log and the ttcn3 TC_mt_ping_pong log, taken with code from branches:
osmo-pcu neels/segv
osmo-ttcn3-hacks neels/pcu

I will continue to look into the life cycle of an UL TBF.
If anyone else has an idea from looking at the logs, help is welcome.

#7 - 08/12/2021 03:19 PM - neels

i notice now that there is a huge branch pespin/fsm in osmo-pcu, it's quite probable that Pau has covered the leak on his branch / that maybe the leak is caused by the branch being partially merged?

#8 - 08/12/2021 03:29 PM - neels

When I test for the UL TBF leak from TC_mt_ping_pong with Pau's branch,
the UL TBF is properly released with "Timeout of X2001".

When I test the 'latest' (v 0.9.0), there is no leak of UL TBF.

So this is a regression in nightly builds, and Pau's work will fix that.
I'll bisect to find the commit that introduces the leak and see if i can get an interim fix applied.

#9 - 08/12/2021 04:10 PM - neels

bisect identifies this commit as the one introducing the UL TBF leak:

```
commit 846fd248dc49c06441da6d7c3cd85df479810f1a
Author: Pau Espin Pedrol <pespin@sysmocom.de>
Date:   Thu Apr 22 20:40:10 2021 +0200

    Stop abusing T3169

    Now that we finally handle N3101 and N3103 correctly, we can fix abuse
    of T3169 we were doing to make sure TBFs were freed.

    According to 3GPP TS 44.060, T3169 should be armed:
    * N3101_MAX reached
    * N3103_MAX reached

    Furthermore, when T3169 is enabled, the tbf should be in state
    RELEASING so that its USF is not used.

    See full description: https://osmocom.org/issues/5033#note-2

    Related: OS#5033
    Change-Id: I2cec531e2633281b88f69ba065c0105580c81076
```

#10 - 08/12/2021 07:32 PM - laforge

On Thu, Aug 12, 2021 at 04:10:34PM +0000, neels [REDMINE] wrote:

> bisect identifies this commit as the one introducing the UL TBF leak:

nice catch! Are you looking to fix that commit, or are you simply going to revert it?

Looking forward to seeing something related in gerrit. Thanks for diving into this.

#11 - 08/13/2021 09:19 AM - neels

I have reverted above commit in https://gerrit.osmocom.org/c/osmo-pcu/+/25193
and the TTCN3 PCU_Tests do pass with this revert applied.
Disclaimer: I'm not really familiar with the PCU and "don't know what I'm doing".

#12 - 08/16/2021 09:16 AM - iedemam

Apologies for not being responsive here. My spam filter was extra greedy it seems.

I see this change has also already been merged. I'll report back if we see the same behavior after upgrades. Otherwise, looks good.

Thanks!

#13 - 09/15/2021 09:26 AM - laforge

- Status changed from In Progress to Feedback

- Assignee changed from neels to pespin

#14 - 09/15/2021 12:38 PM - pespin

- Assignee changed from pespin to iedemam

Same as #5205, let's ask iedemam if he stills sees it with recent versions of osmo-pcu after FSM refactoring was merged (end of August).

Files

pcu-talloc-context.txt	633 KB	08/06/2021	iedemam
osmo-pcu_TC_mt_ping_pong.log	72.7 KB	08/12/2021	neels
_PCU_Tests.TC_mt_ping_pong.merged.gz	3.57 MB	08/12/2021	neels