I noticed, now that LLC block parsing has been improved in wireshark [1], that osmo-pcu tends to send lots of RLCMAC blocks with 2 LLC fragments, specially in CS2 & upwards.

This 2nd LLC data fragment can be identified because it contains:

Frame 33759: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
User Datagram Protocol, Src Port: 59082, Dst Port: 4729
GSM TAP Header, ARFCN: 866 (Downlink), TS: 5, Channel: PDTCH (0)
GPRS DL DATA (CS2)
  DL_Data_Mac_Header
    00... = Payload Type (DL): RLC/MAC block contains an RLC data block (0)
    ..00... = RRBP: Reserved Block: (N+13) mod 2715648 (0)
    .... 0... = S/P: RRBP field is not valid
    .... .111 = USF: 7
    00... = PR: 0 dB (included) to 3 dB (excluded) less than BCCH level - P0 (0)
    ..00 000. = DL TFI: 0
    .... ....0 = FBI: Current Block is not last RLC data block in TBF
    0000 001. = BSN: 1
    .... ....0 = Extension: Extension octet follows immediately
    0111 01.. = Length Indicator: 29
    .... ....1 = ME: no more LLC segments in this RLC block after the current segment, no more extension octets (1)
    .... ....1 = Extension: No extension octet follows
    data segment: LI[0]=29 indicates: (Last segment of) LLC frame (29 octets)
    Data (29 bytes)
    Data: 43c0012b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b
    <!--------- HERE!!!: 43c001 + 2b padding filling what's left of the remaining space
    [Length: 29]
    0000 0000 = DL SPARE: 0

Apparently it sends it when the end of the current LLC PDU is finished in current rclmac DL data block and there's still some space left.

This kind of frame is actually documented, but AFAIU it's being used in a wrong way. It should not be used in this kind of scenarios, but in others explained by the spec.

In this kind of scenarios AFAIU we should be simply sending 1 data block, followed by filling padding until the end.

The use of LLC UI Dummy commands is documented in:

- 3GPP TS 44.060 version 16.0.0 "9.3.1a Delayed release of downlink Temporary Block Flow"
  https://www.etsi.org/deliver/etsi_ts/144000_144099/144060/16.00.00_60/ts_144060v160000p.pdf

- 3GPP TS 44.064 version 16.0.0 "6.4.2.2 Unconfirmed Information (UI) Dummy command"
  https://www.etsi.org/deliver/etsi_ts/144000_144099/144064/16.00.00_60/ts_144064v160000p.pdf

In my understanding, current behavior is not strongly bad, but it doesn't server any purpose sending those since they are simply
ignored by the MS at the LLC layer. It only makes sense to send them if there's really no information to send to the MS but we still want to keep the DL TBF open for a while while we wait for more DL data to come.

[1] https://gitlab.com/wireshark/wireshark/-/merge_requests/842

Associated revisions

Revision aebb77ac - 03/02/2021 01:15 PM - Pau Espin Pedrol

tbf_dl: Fix m_last_dl_drained_fn not set under some conditions

Old commit getting rid of LLC UI dummy and updating create_new_bsn() function introduced a bug by not moving update of value m_last_dl_drained_fn prior to a new break introduced.
As a result, the value is not updated in the case LLC queue becomes drained but last few bytes are drained at exactly that moment.
As a result, the IDLE tbf timer (X2031, keep_open())) returns always true since according to it the drain never happened.

Related: OS#4849
Fixes: 7d0f9a0ec383fcfca934731bd6979b6be6629c90
Change-Id: I7420aeffda3500bcddc990291e44a26511af433ff9

Revision 8afc6bad - 03/02/2021 05:24 PM - Pau Espin Pedrol

tbf_dl: Fix m_last_dl_drained_fn not set under some conditions

Old commit getting rid of LLC UI dummy and updating create_new_bsn() function introduced a bug by not moving update of value m_last_dl_drained_fn prior to a new break introduced.
As a result, the value is not updated in the case LLC queue becomes drained but last few bytes are drained at exactly that moment.
Furthermore, then the IDLE tbf timer (X2031, keep_open())) returns always true since according to it the drain never happened.

The impact of the bug is basically delaying a bit more than expected the time the TBF stays in IDLE state with the TBF release process yet to be started.

Related: OS#4849
Fixes: 7d0f9a0ec383fcfca934731bd6979b6be6629c90
Change-Id: I7420aeffda3500bcddc990291e44a26511af433ff9

History

#1 - 11/13/2020 01:28 PM - pespin

The situation is even worse because counter CTR_RLC_DL_PAYLOAD_BYTES which is supposed to count RLC DL Payload bytes is counting all those forged LLC dummy frames, which means the counter values makes not much sense from someone expecting it to roughly match amount of data received at the Gb side of osmo-pcu.

#2 - 11/27/2020 04:56 PM - pespin

- Status changed from New to In Progress
- % Done changed from 0 to 70

LLC UI dummy frames are not sent anymore together with other llc frames in same rlcmac block thanks to this patch: https://gerrit.osmocom.org/c/osmo-pcu/+/21387 DI TBF: Get rid of LLC UI dummy blocks following other data

#3 - 11/30/2020 07:42 PM - pespin

- Status changed from In Progress to Feedback
- % Done changed from 70 to 90

I submitted TTCN3 test cases to make sure we are not sending them and that we add the padding LI=127 in EGPRS.
https://gerrit.osmocom.org/c/osmo-llc-hacks/+/21419 pcu: Verify LLC UI dummy frames are not appended at end of rlcmac DL block

#4 - 12/09/2020 12:42 PM - pespin

- Status changed from Feedback to Resolved
- % Done changed from 90 to 100

03/08/2022 2/3
Merged, closing ticket.